



HAL
open science

Calibration of a two-step Incremental Sigma-Delta Analog-to-Digital Converter

Li Huang

► **To cite this version:**

Li Huang. Calibration of a two-step Incremental Sigma-Delta Analog-to-Digital Converter. Electronics. Université Paris-Saclay, 2020. English. NNT : 2020UPAST041 . tel-03182829

HAL Id: tel-03182829

<https://theses.hal.science/tel-03182829>

Submitted on 26 Mar 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Calibration of a two-step Incremental Sigma-Delta Analog-to-Digital Converter

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n° 575, Electrical, Optical, Bio: PHYSICS
AND ENGINEERING (EOBE)
Spécialité de doctorat: électronique et optoélectronique, nano-
et microtechnologies
Unité de recherche: Université Paris-Saclay, CentraleSupélec, CNRS,
Laboratoire de Génie Electrique et Electronique de Paris, 91192,
Gif-sur-Yvette, France.
Réfèrent: : CentraleSupélec

**Thèse présentée et soutenue à Gif-sur-Yvette, le 18/12/2020,
par**

LI HUANG

Composition du jury:

Souhil Megherbi Professeur, Centre de Nanosciences et Nanotechnologies, CNRS UMR 9001	Président
Patricia Desgreys Professeure, Télécom Paris	Rapporteuse & Examinatrice
Gilles Sicard Chercheur, HDR, CEA-LETI	Rapporteur & Examineur
Fabrice Guilloux Chercheur, CEA-IRFU	Examineur
Philippe Bénabès Professeur, GeePs, CentraleSupélec	Directeur de thèse
Caroline Lelandais- Perrault Maître de conférence, GeePs, CentraleSupélec	Encadrante & Examinatrice
Anthony Kolar Maître de conférence, GeePs, CentraleSupélec	Encadrant & Examineur



Remerciements

Tout d'abord je voudrais remercier tous les membres du jury venus participer à la soutenance malgré leur emploi du temps chargé, le président du jury Souhil Megherbi, les rapporteurs Patricia Desgreys et Gilles Sicard , ainsi que l'examineur, Fabrice Guilloux. Je tiens à les remercier d'avoir fait des efforts pour comprendre mon travail et d'avoir lancé des discussions scientifiques.

J'aimerais donner mes sincères remerciements à mon directeur de thèse, Philippe Bénabès, et à mes encadrants, Caroline Lelandais- Perrault et Anthony Kolar. Merci à Philippe de m'avoir toujours donné beaucoup de conseils précieux basés sur sa riche expérience et ses connaissances approfondies dans le domaine du $\Sigma\Delta$ au cours de ces trois ans. Je tiens également à remercier Caroline, non seulement pour ses conseils professionnels, mais aussi pour son attention et son souci concernant ma vie. Je tiens également à remercier Anthony pour son humour, ses encouragements et ses conseils philosophiques. Je vous remercie tous pour m'avoir permis d'évoluer sur le plan professionnel et personnel et je considère que j'ai eu beaucoup de chance travailler avec vous.

J'aimerais également remercier tous les collègues dans l'équipe, à savoir Sophie Goncalves pour l'aide administrative, Emilie Avignon, Pietro Maris, Laurent Bourgois et Jérôme Juillard pour les conseils d'enseignement.

J'aimerais également remercier spécialement Francis Trélin. En plus d'exprimer ma gratitude pour le soutien technique, je tiens également à vous remercier de m'avoir amenée à explorer et apprécier la culture et la vie française. J'en garderai de précieux souvenirs.

En plus, je tiens à remercier les amis de mon bureau pour l'ambiance conviviale , Dr Pierre Bisiaux, Dr Ludwig Cron, Dr Juliette Le Hir et Dr Joao Alberto De Franca Ferreira, ainsi que les autres doctorants, Saman Fatima et Joao Roberto Raposo.

J'aimerais aussi remercier les amis de Bourg-La-Reine, Zhang Yihan, Sun Luxi, Quan Yibo et Song Chao, pour leur compagnie pendant la période difficile du confinement. D'ailleurs, un petit merci aussi aux amis de l'équipe de jeu mobile et de l'équipe du Crous.

Pour finir, je voudrais remercier mes parents en Chine qui m'ont toujours soutenu durant toutes

ces années d'études.

Résumé français

La technologie CMOS est de plus en plus utilisée pour réaliser les imageurs. Cette tendance s'accompagne de nouvelles exigences sur les convertisseurs analogiques-numériques (ADC). En particulier, afin d'améliorer la vitesse de conversion, on fait appel à des ADC colonne en parallèle. Aussi, afin de répondre aux besoins d'applications haut de gamme, des ADC à haute résolution sont nécessaires. Typiquement dans le contexte d'imageurs dédiés aux applications d'observation de la Terre, des imageurs Haute Définition (1920×1080) avec des vitesses de 100 Frames par seconde sont requis. Pour un ADC colonne, ces spécifications conduisent à une exigence de l'ordre de 200 kS/s avec une résolution de 14 bit tout en respectant une largeur de 10 μm , largeur typique d'une colonne de pixel. Ainsi, un ADC de type Sigma-Delta incrémental à deux étapes, capable de répondre aux exigences de résolution et de vitesse de conversion a été conçu au sein de notre équipe. Le choix architectural a été guidé par la contrainte de largeur de l'ADC colonne.

Le cœur d'un ADC de type Sigma-Delta ($\Sigma\Delta$) ou Sigma-Delta incrémental ($I\Sigma\Delta$) est un modulateurs Sigma-Delta. Après des décennies de développement, il existe diverses architectures de modulateurs Sigma-Delta. L'ordre des modulateurs Sigma-Delta a été augmenté, du premier ordre jusqu'au quatrième ordre, afin d'obtenir une résolution plus élevée avec le même nombre de cycles d'horloge autrement dit à facteur de sur-échantillonnage (OSR) constant. Cependant, la stabilité n'est pas assurée dans les architectures de modulateurs d'ordre élevé. Afin d'éviter le problème de stabilité du modulateur d'ordre élevé, les architectures des ADC à plusieurs étages sont apparues. En plus des ADC composés purement d'un modulateur Sigma-Delta, il existe également des ADC combinant un modulateur Sigma-Delta avec un ADC d'une autre architecture, dans le but d'augmenter la résolution sans sacrifier la vitesse de conversion. On parle alors d'ADC hybride. Et enfin il existe la solution retenue par notre équipe qui consiste à faire la conversion en deux étapes en utilisant la même architecture pour les deux étapes de conversion, ce qui nous permet d'obtenir 14 bits de résolution avec un OSR de 72.

Toutefois, il est évident que les imperfections du circuit vont dégrader la résolution. Nous

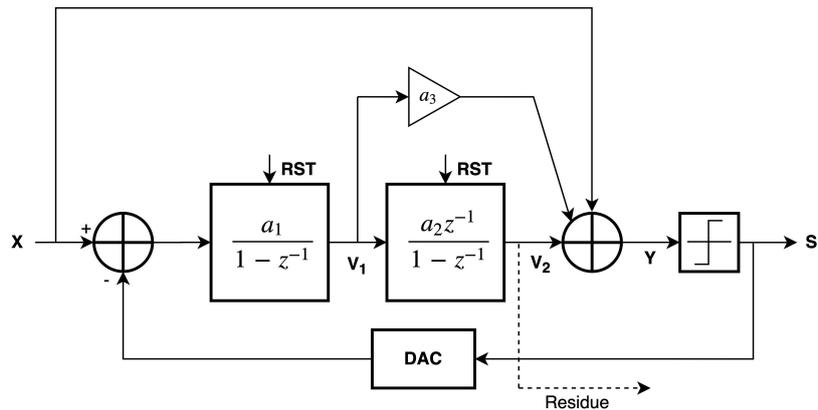


Figure 1: Architecture du modulateur $\Sigma\Delta$ incrémental

envisageons alors d'appliquer une technique de calibration. De manière générale, dans les modulateurs Sigma-Delta, autant les erreurs linéaires sont aisément corrigées, autant les erreurs non-linéaires sont plus difficiles à calibrer. Certaines techniques de linéarisation pour les ADC Sigma-Delta ont été développées pour les blocs d'intégration qui sont les constituants principaux d'un modulateur Sigma-Delta. Malheureusement, il est difficile pour nous de profiter ces méthodes car elles s'appuient sur des architectures incluant un quantificateur multi-bit or notre architecture met en œuvre un quantificateur mono-bit. Plus précisément dans les ADC Sigma-Delta incrémentaux il n'y a pas à notre connaissance de méthode corrigeant spécifiquement les erreurs non-linéaires. Il existe en revanche des techniques de reconstruction numériques reconstruisant au mieux les entrées du convertisseur linéairement à partir des bits de sortie. L'objectif de mon travail est donc d'étudier et proposer de nouvelles techniques de correction pour un ADC Sigma-Delta incrémental. Toutes mes recherches se baseront sur l'architecture et les résultats du circuit réalisé par l'équipe qui s'intitule Pieretta.

Les ADC de type Sigma-Delta incrémental peuvent être considérés comme des ADC de type Sigma-Delta avec une de réinitialisation au début de conversion. Il comprend généralement un modulateur Sigma-Delta et un filtre de reconstruction numérique. L'architecture du modulateur de Pieretta a été choisie par Monsieur Pierre Bisiaux. Comme le montre la Fig. 1, c'est une architecture de modulateur à intégrateur en cascade du second ordre avec feedforward. L'avantage de cette architecture est qu'elle annule le signal d'entrée à l'entrée du filtre de boucle pour qu'il n'y ait que le bruit de quantification à traiter dans la boucle.

La Fig. 2 montre le schéma du circuit analogique de Pieretta. Les transistors dans le circuit ont été dimensionnés par la méthode gm/Id. A noter que les amplificateurs dans les intégrateurs

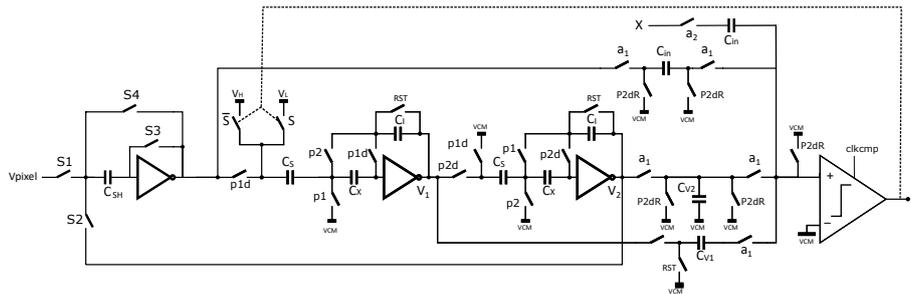


Figure 2: Schéma de la partie analogique du modulateur $\Sigma\Delta$ incrémental

sont des inverseurs avec "gain-boosting" dont le schéma est montré dans la Fig. 3.8. Grâce à l'utilisation des intégrateurs basés sur des inverseurs, la taille du circuit est minimisée par rapport l'utilisation d'intégrateurs conventionnels.

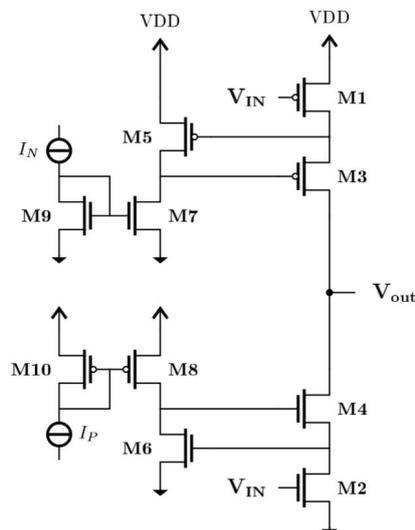


Figure 3: Schéma de l'amplificateur basé sur l'inverseur avec "gain-boosting"

Un circuit numérique non représenté sur le schéma génère les signaux contrôlant les interrupteurs. Ce circuit se compose d'une machine d'état et d'un bloc de non-overlap.

Compte tenu de la nécessité de caractériser le circuit une fois fabriqué, nous avons conçu le circuit pour qu'il soit programmable. Avec des paramètres configurables, nous pouvons faire fonctionner le modulateur $\Sigma\Delta$ dans divers modes de conversion. Cette idée a été réalisée par une machine d'état codée en Verilog. La machine d'état génère des signaux d'activation en fonction des réglages des paramètres. Le bloc non-overlap reçoit les signaux d'activation de la machine d'état puis génère les signaux de commande avec les retards appropriés vers le circuit analogique afin d'assurer le bon séquençage des activations d'interrupteurs.

Le circuit a été conçu avec la technologie de 180 nm de XFAB. La simulation post-layout du

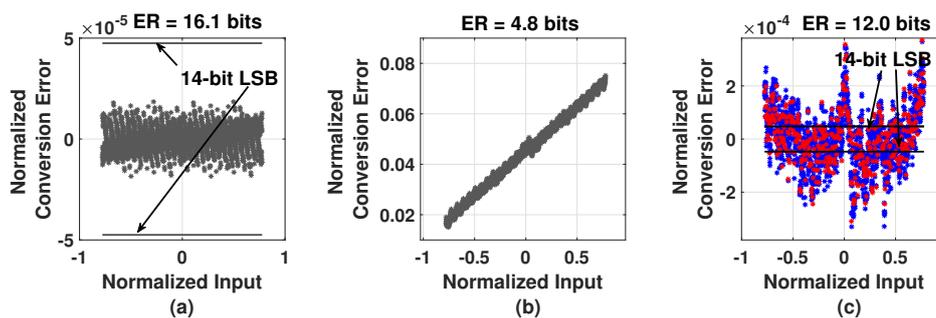


Figure 4: Résultats de simulation: (a) du modèle idéal; (b) en post-layout, avec le filtre théorique; (c) en post-layout, avec la correction de gain et d'offset.

modulateur en deux étapes a mis en évidence une forte dégradation de la résolution, comme le montre la Fig. 4b, par rapport à modèle idéal dont l'erreur de quantification est montrée dans la Fig. 4a. Après une correction de gain et d'offset, nous obtenons une résolution équivalente (ER) de 9.8 bits, comme le montre la Fig. 4c, à noter que les points en bleu servent à estimer des paramètres et ceux en rouge servent à vérifier. ER ne correspondant pas à notre résolution cible de 14 bits, une correction est donc nécessaire.

Notre but a d'abord été de trouver un algorithme pour reconstruire l'entrée du modulateur à partir des combinaisons de bits des sorties du modulateur en considérant le modulateur comme un boîte noire. Un tel algorithme sera implémenté dans un filtre numérique de reconstruction.

En plus de la correction de gain et d'offset, la méthode de correction habituelle est un filtre optimal dont le principe est de pondérer les coefficients de chaque bit ainsi que l'offset. La Fig. 5 montre les résultats corrigés où le filtre optimal n'est appliqué que sur les bits issus de la première étape. Même si l'ER atteint notre but, nous cherchons encore à améliorer la résolution sans aucune restriction sur la complexité de l'algorithme dans un premier temps.

Après avoir analysé les signaux dans la simulation en post-layout, nous avons trouvé que les erreurs de certains signaux internes étaient corrélées aux séquences des bits de sortie du modulateur. C'est pour quoi nous proposons une méthode de correction qui reconstruit l'entrée du modulateur en fonction de la séquence des bits. Nous rappelons cette méthode: filtre de pattern-correcting. Le principe de cet algorithme est de détecter la combinaison (ou motif) de paquets de 3 bits consécutifs dans la séquence de bits de sortie du modulateur. La Fig. 6 illustre technique de détection. Huit nouveaux vecteurs c_{bi} , ($i = 1, 2, \dots, 8$) sont créés. En balayant la séquence des bits de sortie S par paquets de 3 bits, on va activer le bit à la position courante dans le vecteur c_{bi} correspondant au pattern détecté. Après avoir scanné tous les paquets, les séquences de bits c_{bi}

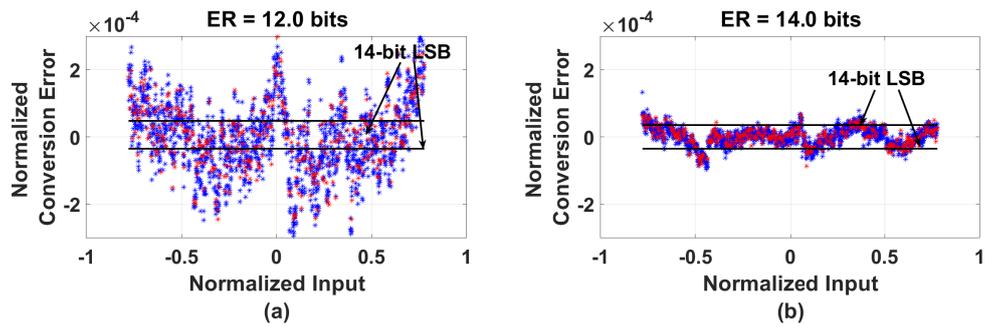


Figure 5: Comparaison des résultats corrigés par les méthodes existées basées sur la simulation en post-layout: (a) avec la correction de gain et d'offset; (b) avec le filtre optimal pour la première étape.

sont filtrées par des filtres à réponse impulsionnelle finie (FIR) et recombinaées comme le montre la Fig. 7. On n'applique le filtre de pattern-correcting que pour la première étape, car la correction de la deuxième étape améliore très peu la résolution si l'erreur résiduelle dans la première étape reste importante. Enfin, l'entrée de modulateur est reconstruite à partir de la sortie du filtre de pattern-correcting et de la double intégration de la deuxième étape et d'un offset.

S	1	-1	-1	1	-1	1	-1	1
CB1 (111)	0	0	0	0				
CB2 (1-11)	0	0	0	1				
CB3 (1-1-1)	1	0	0	0				
CB4 (11-1)	0	0	0	0				
CB5 (-11-1)	0	0	1	0				
CB6 (-111)	0	0	0	0				
CB7 (-1-11)	0	1	0	0				
CB8 (-1-1-1)	0	0	0	0				

Figure 6: Illustration de détection de pattern

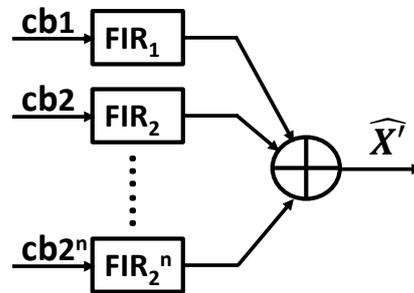


Figure 7: Algorithme du filtre de pattern-correcting

Les résultats après la correction du filtre de pattern-correcting dans la simulation en post-layout sont montrés dans la Fig. 8. Nous observons une amélioration de la résolution jusqu'à 15 bits.

Comme dit précédemment, une résolution de 14 bits a été obtenue en appliquant le filtre optimal, ce qui nécessite 38 coefficients. Mais dans notre contexte, parce que nous gérons des ADC colonne pour un imageur HD, le nombre total de coefficients risque d'être inacceptable. Afin de réduire le nombre de coefficients tout en ciblant 14 bits, nous proposons de simplifier la méthode de correction basée sur la correction du filtre optimal. Comparé à un FIR, un filtre

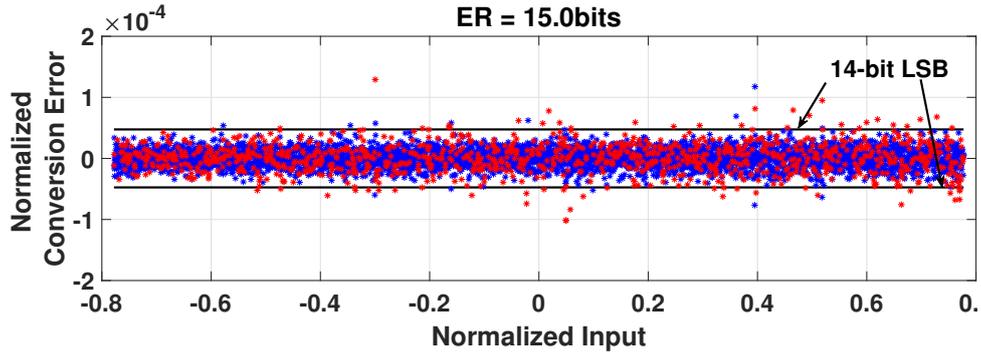


Figure 8: Résultat de la correction avec un filtre de pattern-correcting

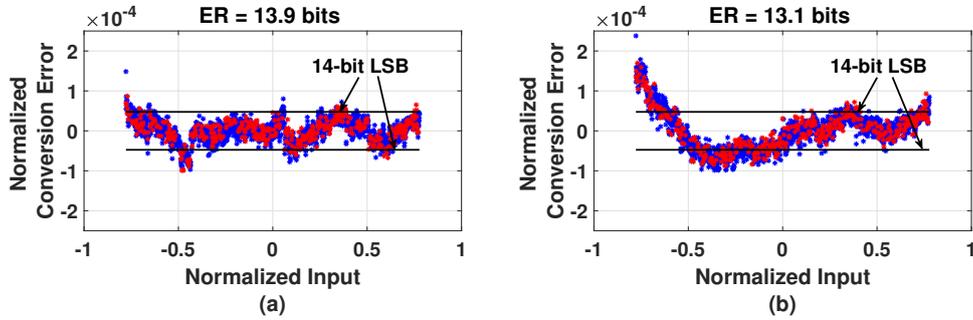


Figure 9: Comparaison des résultats corrigés par : (a) le filtre optimal ; (b) le filtre simplifié

à réponse impulsionnelle infinie (IIR) nécessite moins de coefficients. Pour cette raison, nous essayons de trouver un IIR dont la réponse impulsionnelle s'approche de celle de notre filtre FIR. Dans ce cas-là, l'entrée du modulateur peut être estimée comme suit:

$$\widehat{V_{pixel,si}} = D_{si,1} + D_{go,2} + O_{si} \quad (1)$$

où $D_{si,1}$ est la sortie de filtre simplifié pour la première étape, $D_{go,2}$ est la double intégration des sorties dans la deuxième étape et O_{si} est le terme pour corriger l'offset.

La comparaison des résultats des corrections avec le filtre optimal et le filtre optimal simplifié dans la simulation post-layout est représentée sur la Fig. 9. L'ER après la correction de filtre optimal simplifié est proche de 13,1 bits et donc un peu moins bon qu'avec le filtre optimal. Mais le filtre simplifié requiert 8 coefficients au lieu de 38 pour le filtre optimal. Puisque le filtre optimal simplifié est une approximation d'un filtre optimal, une dégradation en ER est normale. Afin de compenser cette perte de résolution, un filtre de correction de motif de paquet de 3 bits ($n = 3$) est utilisé pour corriger les combinaisons des premiers 3 MSBs de la sortie du modulateur dans

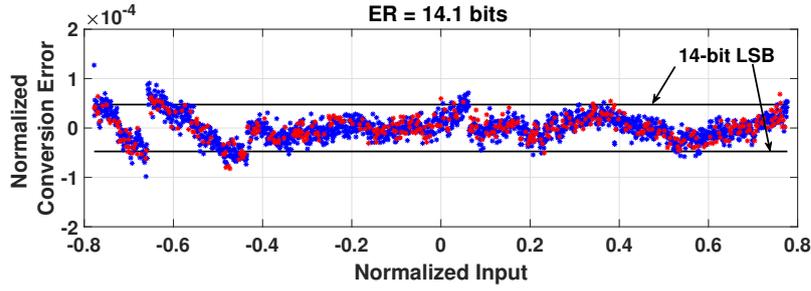


Figure 10: Résultat corrigé par le filtre hybride

la première étape. En combinant la correction du filtre optimal simplifié et la correction du filtre pattern-correcting, l'entrée de l'ADC peut être estimée comme suit:

$$\widehat{V_{pixel,hy}} = D_{si,1} + D_{pa3,mbit} + D_{go,2} + O_{hy} \quad (2)$$

où $D_{si,1}$ est la sortie du filtre optimal simplifié de la première étape, $D_{pa3,mbit} = \sum_{i=1}^8 \mathbf{c}_{bi} \cdot \mathbf{fi}$ est la somme des sorties de FIR_i correspondant, $D_{go,2}$ est la double intégration des sorties dans la deuxième étape et O_{hy} est le terme pour corriger l'offset. On appelle ce filtre, filtre hybride.

Les résultats de correction avec le filtre hybride dans la simulation post-layout sont présentés sur la Fig. 10. La résolution cible est atteinte et cette méthode nécessite 16 coefficients. Cette complexité est plus acceptable pour notre ADC colonne.

Bien que nous ayons réussi à corriger la sortie de l'ADC et à obtenir la résolution cible, nous souhaitons tout de même comprendre la raison de la dégradation dans la simulation post-layout. Pour cet objectif, une simulation en post-layout a d'abord été effectuée où les valeurs de tous les signaux internes à chaque cycle d'horloge ont été sauvegardées. En analysant les comportements de certains signaux, nous proposons un modèle au niveau circuit pour les intégrateurs à capacités commutées basés sur un inverseur impliquant des capacités parasites raisonnables à notre avis. Ces capacités parasites ont été identifiées et estimées selon certains signaux internes aux intégrateurs dans des simulations post-layout. Le schéma de l'intégrateur avec des capacités parasites identifiées est présenté dans la Fig. 11. Ces modèles d'intégrateur permettent de simuler les intégrateurs dans la simulation en post-layout avec une erreur quadratique moyenne de $10^{-4}V$.

Au cours de la troisième année de ma thèse, nous avons reçu la puce de circuit fabriqué. Selon les résultats des tests, le circuit réel est pollué exceptionnellement par un bruit d'une puissance élevée anormale. Le bruit cause des problèmes dans le test à la fois lors de l'échantillonnage

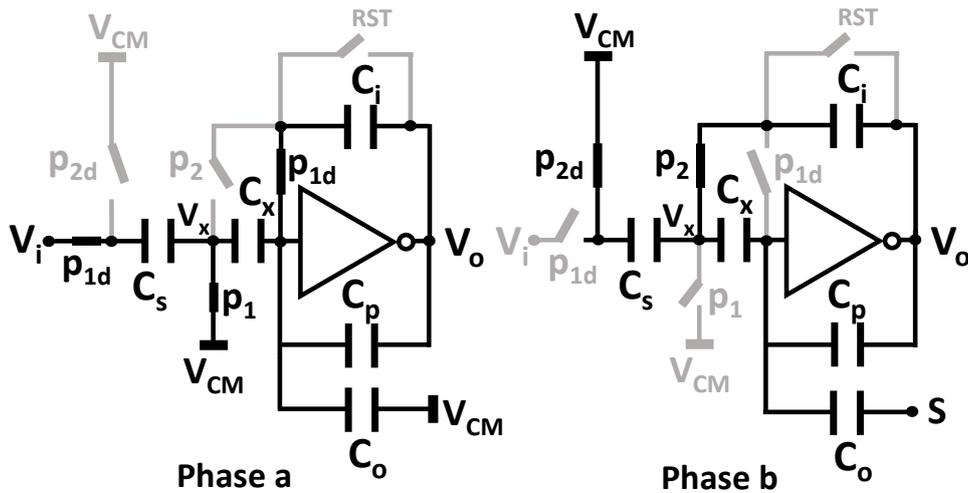


Figure 11: Schéma au niveau de circuit d'un intégrateur basé sur inverseur avec capacités parasites dans la phase d'échantillonnage (Phase a) et dans la phase d'intégration (Phase b).

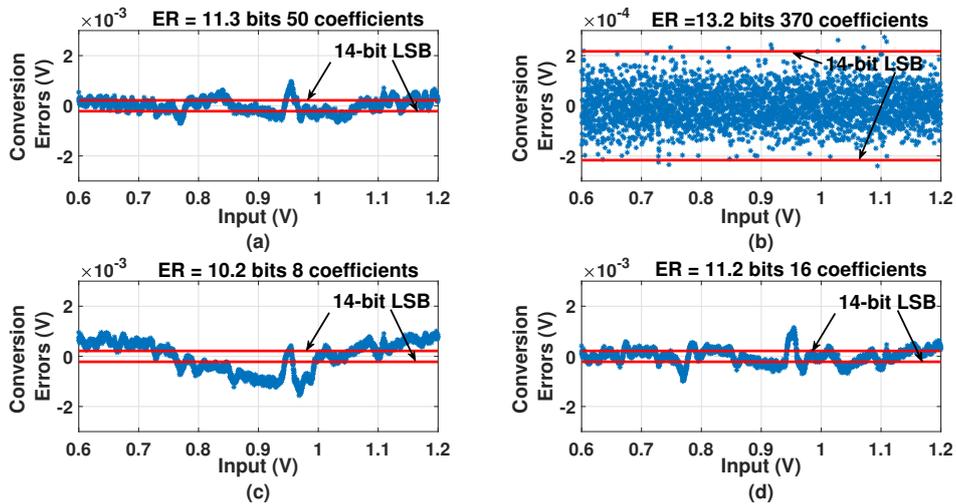


Figure 12: Résultats de mesure après corrections: (a) filtre optimal; (b) filtre pattern-correcting; (c) filtre simplifié; (d) filtre hybride.

et lors du post-traitement. Toutefois les résultats de mesure filtrés montrent qu'indépendamment d'une dégradation globale de l' ER , comme le montre dans la Fig. 12 le filtre optimal et toutes les méthodes de correction proposées peuvent améliorer la résolution du circuit réel. La dégradation résiduelle est sans doute causée par le bruit restant ou les capacités parasites.

En conclusion, cette thèse se concentre sur la calibration et la correction d'un ADC $\Sigma\Delta$ incrémental en deux étapes et propose de nouvelles méthodes de correction pour compenser des défauts de réalisation. L'effet de ces méthodes a été validé par des simulations post-layout même aussi sur un circuit réel. Un nouveau modèle d'intégrateur basé sur un inverseur a aussi été proposé. Ces modèles permettent d'évaluer la sensibilité de l'ADC après correction aux valeurs des capacités parasites. En identifiant ainsi la ou les capacités parasites les plus critiques, on peut

ainsi optimiser plus efficacement un futur design.

Contents

1 Introduction	13
1.1 From CMOS Image Sensor to Column-parallel ADC	13
1.2 Challenge of ADC in the Earth Observation Application	17
1.3 Pieretta: Chip of New Proposed Two-step I $\Sigma\Delta$ ADCs	17
1.4 Motivation and Problematic	20
1.5 Thesis Organization	23
2 State of the art	27
2.1 Incremental $\Sigma\Delta$ ADC	27
2.2 Linearization Techniques for $\Sigma\Delta$ ADCs	32
2.3 Researches on I $\Sigma\Delta$ ADC Reconstruction Filter	39
2.4 Conclusion	42
3 Two-Step I$\Sigma\Delta$ ADC and Circuit Pieretta	43
3.1 I $\Sigma\Delta$ ADC Modeling	43
3.1.1 Second-Order I $\Sigma\Delta$ Modulator	45
3.1.2 Two-step conversion	47
3.1.3 Digital Reconstruction Filter	49
3.2 Design of I $\Sigma\Delta$ Modulator	50
3.2.1 Analog Circuit Part	50
3.2.2 Digital Circuit	57
3.3 Post-layout simulation results	60
3.3.1 Equivalent resolution	61
3.3.2 Parallel Simulations and Choice of the Modulator Input Values	62
3.3.3 Post-layout Simulation of One-Step Conversion	62
3.3.4 Post-layout Simulation of Two-Step Conversion	64

3.4 Conclusion	65
4 I$\Sigma$$\Delta$ ADC correction technique	67
4.1 Global error mode	68
4.2 Correction methodology	70
4.3 State-of-the-art correction methods	70
4.3.1 Correction of gain and offset	71
4.3.2 Optimal filter	72
4.3.3 Correction results of existing methods	73
4.4 New proposed correction methods	77
4.4.1 Pattern-correcting filter	78
4.4.2 Correction results of pattern-correcting filter techniques applied to post-layout simulation	79
4.5 Corrections results for multi-step conversions	82
4.6 Correction method simplification	83
4.6.1 Simplified Filter	84
4.6.2 Hybrid filter	85
4.6.3 Simplified hybrid correction	86
4.7 Coefficient quantization	87
4.8 Resolution optimization for two-step conversion	88
4.9 Conclusion	89
5 Circuit defects modeling	91
5.1 Classical circuit non-idealities	92
5.1.1 Non-linear amplifier gain	92
5.1.2 Capacitor mismatch	94
5.1.3 Simulation of the modulator model with classic defects	94
5.2 New inverter-based SC integrator model with parasitic capacitors	96
5.2.1 Abnormal Variation of Certain Intern Signals in Integrators	96
5.2.2 Discovery of parasitic capacitors	96
5.2.3 Circuit-level models of integrators	98
5.3 Validation of the new proposed inverter-based SC integrator models	102
5.3.1 Parameters identification for the new proposed inverter-based SC integrator models	102

5.3.2 Comparison of the new proposed inverter-based SC integrator models to the ideal integrator model	102
5.4 Validation of the $\Sigma\Delta$ modulator model composed of the new proposed inverter-based SC integrator models	103
5.4.1 Parameters identification of other imperfections in $\Sigma\Delta$ modulator	105
5.4.2 Comparison of proposed modulator model and modulator in post-layout simulation	105
5.4.3 Study of the parasitic capacitor impacts on correction methods	111
5.4.4 Study of the noise resistance of correction methods	114
5.5 Further study on impact of parasitic capacitor to an inverter-based $\Sigma\Delta$ modulator	116
5.5.1 Deduction of transfer function of the 2 nd $\Sigma\Delta$ modulator	116
5.5.2 Use of actual filter as the reconstruction filter for the non-ideal modulator	119
5.5.3 Error analysis	120
5.6 Conclusion	120
6 Measurement	123
6.1 Package of Pieretta	123
6.2 Equipment of Test-bench	125
6.3 Asynchronous Sampling	126
6.4 Measurement Results	127
6.4.1 Interference of Noise and Primary Measurement Results	127
6.4.2 Filtered Measurement Results	130
6.5 Conclusion	131
7 Conclusion	133
A Study of Convergence in Parameter Estimation	135

List of Figures

1	Architecture du modulateur $\Sigma\Delta$ incrémental	6
2	Schéma de la partie analogique du modulateur $\Sigma\Delta$ incrémental	7
3	Schéma de l'amplificateur basé sur l'inverseur avec "gain-boosting"	7
4	Résultats de simulation: (a) du modèle idéal; (b) en post-layout, avec le filtre théorique; (c) en post-layout, avec la correction de gain et d'offset	8
5	Comparaison des résultats corrigés par les méthodes existées basées sur la simulation en post-layout: (a) avec la correction de gain et d'offset; (b) avec le filtre optimal pour la première étape	9
6	Illustration de détection de pattern	9
7	Algorithme du filtre de pattern-correcting	9
8	Résultat de la correction avec un filtre de pattern-correcting	10
9	Comparaison des résultats corrigés par : (a) le filtre optimal ; (b) le filtre simplifié	10
10	Résultat corrigé par le filtre hybride	11
11	Schéma au niveau de circuit d'un intégrateur basé sur inverseur avec capacités parasites dans la phase d'échantillonnage (Phase a) et dans la phase d'intégration (Phase b)	12
12	Résultats de mesure après corrections: (a) filtre optimal; (b) filtre pattern-correcting; (c) filtre simplifié; (d) filtre hybride	12
1.1	CMOS image sensor applied in digital camera	14
1.2	CMOS image sensor architecture	15
1.3	Pepline ADC architecture	16
1.4	Multi-step ADC architecture	17
1.5	Pixel Matrix with two columns of ADCs in parallel	17
1.6	Architecture of a second-order $I\Sigma\Delta$ modulator	18

1.7	Quantization errors of the $\Sigma\Delta$ modulator in the function of the input amplitude when 25 clock cycles are applied.	19
1.8	Equivalent number of bits of the $\Sigma\Delta$ modulator in the function of the input amplitude for some values of OSR .	19
1.9	Relatively occurrence frequency of the residues of each steps when 200 clock cycles are applied.	20
1.10	Diagrams of inverter-based amplifiers: (a) Simple inverter; (b) Cascaded inverter; (c) Cascaded inverter with gain boosting.	21
1.11	A typical op-amp's DCG versus output voltage with the rail-to-rail voltage of V_{dd} .	23
1.12	The FDCG and DCG linearity required for the output SNDR nearly 90dB.	24
1.13	The PSDs of the modulator output with the first integrator nonlinear DCG of 9, 18, and 37 dB.	24
2.1	A/D converter based on PCM	28
2.2	A basic sigma-delta A/D converter implemented with bipolar	29
2.3	Uni-polar first-order incremental sigma-delta A/D converter	30
2.4	High-order $\Sigma\Delta$ ADC architecture	31
2.5	MASH 1-1 $\Sigma\Delta$ ADC architecture	31
2.6	Architecture of two-step pipeline $\Sigma\Delta$ ADC	32
2.7	Hybrid SAR- $\Sigma\Delta$ ADC architecture	32
2.8	Architecture of a Extended Counting $\Sigma\Delta$ ADC in $\Sigma\Delta$ conversion	33
2.9	Topology of a second-order $\Sigma\Delta$ modulator	33
2.10	Polynomial model of the output-signal-dependent amplifier gain	34
2.11	Equivalent model of a non-ideal first-order modulator	35
2.12	Equivalent model of a non-ideal second-order modulator	35
2.13	Proposed adaptive digital calibration schematic of a first-order $\Sigma\Delta$ modulator: (a) Architecture, (b) error filter (H_e), and (c) LMS update block	36
2.14	MASH 2-2 $\Sigma\Delta$ modulator.	37
2.15	Block diagram of (a) integrator output approximation, (b) modulator correction, (c) error estimation, and (d) error parameter identification.	37
2.16	MASH 2-0 ADC	38
2.17	(a) MASH 2-0 ADC with digital noise-leakage compensation, (b) scheme of the digital noise-leakage compensation filter and the correlator.	38
2.18	ADC in calibration mode: parallel LMS coefficient estimation	39

2.19 Improved first-order converter with second-order filtering and dither.	40
2.20 Quantization error of a 10-bit first-order converter around zero input. (a) Using first-order digital filter. (b) Using second-order digital filter (two integrators). (c) Using second-order filter with dither signal injected into the loop	41
2.21 Block diagram of a typical $\Sigma\Delta$ ADC with thermal noise at the input.	42
3.1 High level architecture of an $\Sigma\Delta$ ADC	44
3.2 A second-order CIFF modulator	45
3.3 Numbers of clock cycles required in the function of ENOB	47
3.4 Comparison of numbers of clock cycles required in one-step and two-step conversions	49
3.5 Schematic of the $\Sigma\Delta$ modulator	51
3.6 Timing diagram of the command signals and corresponding states of components	51
3.7 Schematic of inverter-based SC integrator	52
3.8 Schematic of gain-boosting inverter-based amplifier	53
3.9 Schematic of Sampling-and-hold block	55
3.10 Timing diagram of the signals commanding S/H block	55
3.11 Schematic of adder-and-comparator	56
3.12 Timing diagram of command signals in adder and comparator	56
3.13 Schematic of double-tail comparator	56
3.14 Schematic of non-overlap block	57
3.15 Examples of timing in finite mode and infinite mode	59
3.16 Diagram of states	60
3.17 Diagram of the inputs generation.	63
3.18 Inputs generated with new method.	64
3.19 Equivalent resolution in one-step mode using the theoretical filter: (a) for the model of ideal ADC; (b) in post-layout simulation; (c) in post-layout simulation after the correction of gain and offset.	64
3.20 Equivalent resolution in two-step mode using the theoretical filter: (a) for the ideal ADC model; (b) in post-layout simulation; (c) in post-layout simulation after the correction of gain and offset.	65
4.1 Diagram of calibration	67
4.2 Diagram of correction methodology	70
4.3 Diagram of the correction of gain and offset.	71

4.4	Diagram of the optimal filter correction.	72
4.5	Correction results in the one-step mode of an ideal $\Sigma\Delta$ modulator : (a) with the correction of gain and offset; (b) with the optimal filter correction	73
4.6	Correction results in the two-step mode of an ideal $\Sigma\Delta$ modulator: (a) with the correction of gain and offset; (b) with the optimal filter correction in plan B	74
4.7	Correction results in the one-step mode in post-layout simulation: (a) with the correction of gain and offset; (b) with the optimal filter correction	75
4.8	Impulse responses of the filters of existing methods for one-step conversion.	76
4.9	Correction results in the two-step mode: (a) with the correction of gain and offset; (b) with the optimal filter correction in plan B.	77
4.10	Impulse responses of the filters of existing methods for two-step conversion.	77
4.11	Errors in S/H block	78
4.12	Illustration of pattern identification	79
4.13	Algorithm of Pattern-correcting filter	79
4.14	Correction result in one-step mode with the pattern-correcting filter correction, $n = 3$	80
4.15	Impulse responses of the pattern-correcting filters for one-step conversion.	80
4.16	Impulse responses of the pattern-correcting filters for two-step conversion.	82
4.17	Diagram of correction methods for two-step conversion after simplification	84
4.18	Comparison of correction results in the two-step mode: (a) with the optimal filter correction in plan A; (b) with the simplified optimal filter correction in plan A	85
4.19	Comparison of the impulse responses in the optimal filter and the simplified filter.	85
4.20	Correction result in the two-step mode with the hybrid filter correction in plan A	86
4.21	Correction result in the two-step mode with the simplified hybrid filter correction in plan A	87
4.22	Comparison of the simplified hybrid filter and the hybrid filter in plan A with different m	87
4.23	Summary of correction methods.	89
4.24	With the optimal filter correction in plan B, global equivalent resolution in two-step mode in the function of M_1	89
4.25	Conversion error when $M_1 = 48$ using simplified hybrid correction in plan A with $n = 3, m = 3$	90
5.2	ER in the function of A_{max}	95
5.3	ER in the function of capacitor mismatch ratios: (a) C_s/C_i ; (b) C_x/C_i	95

5.4	$V_{x,b}$ in the function of the integrator outputs $V_{o,b}$	96
5.5	$V_{x,b}$ in the function of difference between the integrator outputs in two phases $V_{o,b} - V_{o,a}$	97
5.6	An inverter-based SC integrator with parasitic capacitors	97
5.7	Comparison of the estimation errors in the first integrator respectively of the ideal integrator model and of the new proposed model.	104
5.8	Comparison of the estimation errors in the second integrator respectively of the ideal integrator model and of the new proposed model.	104
5.9	Conversion errors in one-step mode.	106
5.10	Conversion error in one-step mode in functional model simulations with non-linear finite amplifier gain: (a) with theoretical filter; (b) with correction of gain and offset; (c) with optimal filter correction; (d) with pattern-correcting filter correction of 3-bit package ($n = 3$); (e) with simplified filter; (f) with hybrid filter of 3-bit package, $m = 3$	108
5.11	Conversion errors in two-step mode in post-layout simulation: (a) with theoretical filter; (b) with correction of gain and offset; (c) with optimal filter correction in plan A; (d) with optimal filter correction in plan A; (e) with pattern-correcting filter correction of 3-bit package ($n = 3$) in plan A; (f) with pattern-correcting filter correction of 3-bit package ($n = 3$) in plan B; (g) with simplified filter in plan A; (h) with hybrid filter with $n = 3, m = 3$ in plan A.	109
5.12	Conversion errors in two-step mode in functional model simulations with infinite amplifier gain: (a) with theoretical filter; (b) with correction of gain and offset; (c) with optimal filter correction in plan A; (d) with optimal filter correction in plan A; (e) with pattern-correcting filter correction of 3-bit package ($n = 3$) in plan A; (f) with pattern-correcting filter correction of 3-bit package ($n = 3$) in plan B; (g) with simplified filter in plan A; (h) with hybrid filter with $n = 3, m = 3$ in plan A.	109
5.13	Conversion errors in two-step mode in functional model simulations with non-linear finite amplifier gain: (a) with theoretical filter; (b) with correction of gain and offset; (c) with optimal filter correction in plan A; (d) with optimal filter correction in plan A; (e) with pattern-correcting filter correction of 3-bit package ($n = 3$) in plan A; (f) with pattern-correcting filter correction of 3-bit package ($n = 3$) in plan B; (g) with simplified filter in plan A; (h) with hybrid filter with $n = 3, m = 3$ in plan A.	110
5.14	Equivalent resolution in one-step mode respectively with the optimal filter correction and the patten-correcting filter correction (package of 3 bits): (a) sweep of C_{p1} and C_{p2} and $C_{o1} = C_{o2} = 0$; (b) sweep of C_{o1} and C_{o2} and $C_{p1}/C_{x1} = C_{p2}/C_{x2} = 0.01$;	112

5.15 Equivalent resolution in one-step mode respectively with theoretical filter and with the optimal filter correction and the simplified filter correction (package of 3 bits): (a) sweep of C_{p1} and C_{p2} and $C_{o1} = C_{o2} = 0$; (b) sweep of C_{o1} and C_{o2} and $C_{p1}/C_{x1} = C_{p2}/C_{x2} = 0.01$;	112
5.16 Equivalent resolution in two-step mode respectively with the optimal filter correction and the patten-correcting filter correction (package of 3 bits): (a) sweep of C_{p1} and C_{p2} and $C_{o1} = C_{o2} = 0$; (b) sweep of C_{o1} and C_{o2} and $C_{p1}/C_{x1} = C_{p2}/C_{x2} = 0.01$;	113
5.17 Equivalent resolution in two-step mode respectively with theoretical filter and with the optimal filter correction and the simplified filter correction (package of 3 bits): (a) sweep of C_{p1} and C_{p2} and $C_{o1} = C_{o2} = 0$; (b) sweep of C_{o1} and C_{o2} and $C_{p1}/C_{x1} = C_{p2}/C_{x2} = 0.01$;	113
5.18 Diagram of the $I\Sigma\Delta$ modulator with noise sources.	114
5.19 Equivalent resolution in the function of noise amplitude in one-step mode: (a) only with noise source N_1 ; (b) only with noise source N_2 ; (c) only with noise source N_y ; (d) with noise sources $N_1 + N_2 + N_y$	115
5.20 Equivalent resolution in the function of noise amplitude in two-step mode: (a) only with noise source N_1 ; (b) only with noise source N_2 ; (c) only with noise source N_y ; (d) with noise sources $N_1 + N_2 + N_y$	115
5.21 ER in the function of C_p applying different reconstruction filters in the simulation of the proposed $I\Sigma\Delta$ modulator model	120
6.1 Layout of the $I\Sigma\Delta$ modulator	124
6.2 Layout of the core of Pieretta	124
6.3 Schematic of test-bench	126
6.4 Timing diagram of clk_out and the modulator output	127
6.5 Timing diagram of signals in asynchronous sampling mode	128
6.6 Generation of clk_out' and the voting mechanism for deciding the values of BS	128
6.7 Comparison between the first 8 MSBs in outputs of different modulators for the same input	129
6.8 Comparison between the modulator outputs and the modulator inputs	129
6.9 Original conversion errors of the real circuits	130
6.10 Filter before correction	131
6.11 Conversion errors after correction, choice 1: (a) optimal filter in plan A; (b) pattern-correcting filter in plan A, $m = 3$; (c) simplified filter; (d) hybrid filter, $m = 3$, $m_{bit} = 3$;	131

6.12 Tendency of ER in two-step simulations.	132
A.1 Differences of the values estimated of the coefficients correcting gain and offset in the function of the numbers of samples used for coefficient estimation.	136
A.2 Differences of the values estimated of the coefficients correcting offset and D_2 in the optimal in the function of the numbers of samples used for coefficient estimation.	137
A.3 Differences of the values estimated of the coefficients of the first three MSBs in the optimal in the function of the numbers of samples used for coefficient estimation.	137
A.4 Differences of the values estimated of the coefficients correcting offset and D_2 in the simplified filter in the function of the numbers of samples used for coefficient estimation.	138
A.5 Differences of the values estimated of the coefficients in IIR filter in the simplified filter in the function of the numbers of samples used for coefficient estimation.	138
A.6 Differences of the values estimated of the coefficients of certain patterns in the hybrid filter in the function of the numbers of samples used for coefficient estimation.	139

List of Tables

1.1 Summary of performances of different ADCs.	16
1.2 Summary of inverter-based amplifiers designed by Pierre.	21
1.3 Summary of the $\Sigma\Delta$ modulator.	22
3.1 Dimensions of transistors in the gain-boosting inverter	54
3.2 Dimensions of transistors in the comparator	56
3.3 Dimensions of transistors in switch	57
3.4 Descriptions of conversion modes with different configurations	60
3.5 Descriptions of states	61
3.6 Descriptions of transfer conditions	61
4.1 ERs applying the existing correction methods in the simulation of the ideal $\Sigma\Delta$ ADC	74
4.2 ERs applying the existing correction methods in post-layout simulation	76
4.3 Comparison of the pattern-correcting filters in one-step mode based on post-layout simulation	81
4.4 Comparison of the pattern-correcting filters in two-step mode based on post-layout simulation	81
4.5 ERs after correction of gain and offset in multi-step modes.	83
4.6 ERs corrected with the optimal filter in plan B of multi-step modes.	83
4.7 ERs corrected with the pattern-correcting filter in plan B of multi-step modes.	83
4.8 Numbers of bits to code a coefficient in the FIRs of different corrections	88
4.9 Numbers of bits to code a coefficient in simplified optimal filter (type IIR)	88
4.10 Comparison of some correction methods in two-step mode with quantized coefficients	88
4.11 Comparison of ER when $M_1 = 36$ and $M_1 = 48$, using simplified hybrid correction in plan A.	89

5.1 Comparison of correction results of the proposed model in two-step mode to post-	
layout simulation	108
6.1 Descriptions of conversion modes with different configurations	125

Chapter 1

Introduction

Complementary metal-oxide-semiconductor (CMOS) image sensor technology has been well developed during the past few decades. The requirements to analog-to-digital converter (ADC) come along with the improvements of CMOS image sensor performance. In order to improve the conversion rate, column-parallel ADCs are widely used in CMOS image sensor applications.

The use of column-parallel ADCs only increases conversion rate. However, for many high-end image sensor applications which are supposed to provide high dynamic range, ADCs of high resolution are required. Apart of this, column-parallel ADC has special constraint on its size: the width should not be too large so that the column-parallel ADCs in ADC array can align with pixels.

Aiming at CMOS image sensors for the Earth observation which needs high-resolution ADCs, a two-step incremental $\Sigma\Delta$ ADC was designed by our team which is able to answer the requirements of resolution and conversion speed. To respect the constraint of size, inverter-based amplifiers have been chosen when the ADC was designed. The use of inverter-based amplifiers might lead to resolution degradation. For the purpose of steadying the ADC resolution, a calibration technique is required. Therefore, here is my PhD subject: development of new calibration technique for a two-step incremental $\Sigma\Delta$ ADC.

1.1 From CMOS Image Sensor to Column-parallel ADC

The first devices with CMOS image sensor (CIS) technology on the market for the public use raised in 1990s. Despite of its low image quality in early years, CIS technology survived in the game thanks to following facts: capability to integrate peripheral functional circuit on the same die as the image sensor; less size and lower power consumption. In a past few decades, CIS technology has significant improvement by overcoming these draw-backs: increased noise and

reduced sensitivity which makes them offer picture quality that meets or exceeds the capabilities of Charge-coupled devices (CCDs) [1]. Today CISs are suitable for many consumer domains (webcams, cell phones, digital cameras...). Fig. 1.2[2] shows a classical CMOS active pixel sensor

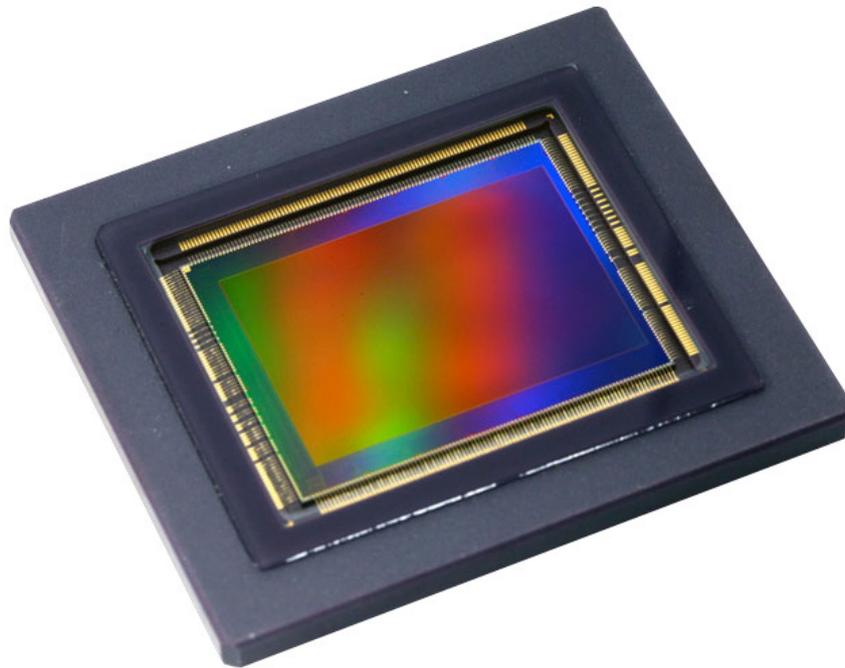


Figure 1.1: CMOS image sensor applied in digital camera

(APS) architecture. An APS consists of 4 main parts: active pixel array, analog signal processors, analog-to-digital converter (ADC) and peripheral logic control circuits. The active pixels are responsible for converting the number of photons into a voltage, via photodiodes. Analog Signal Processors perform functions such as charge integration, gain sample and hold, correlated double sampling (CDS) and FPN suppression [3]. ADCs, as the name, convert analog signals to digital so that they can be used for the following signal processing.

After market revolution at the beginning of 2000, HD (high definition) is becoming the standard all around the world which means increased number of pixels in image sensor and decreased pixel array readout speed. To ensure frame rate, single ADC covering all pixels should work in very high speed which results in undesirably high power consumption. In order to release the constraint of ADC conversion frequency, the column-parallel ADC architecture was proposed.

Amount the existing column-parallel ADC dedicated to CMOS image sensor, single-slope ADC (SS ADC) are friendly in size. [4][5] provide a conversion rate higher than 100 kS/s and a maximum resolution of 10-11 bits. [6] Uses hybrid column counters and achieves 12 bits. Although it is possible to modify to the ADC in [6] in order to get higher resolution, the clock frequency of the counter in the ADC should be very high. Cyclic ADCs [7][8] which are compatible in size

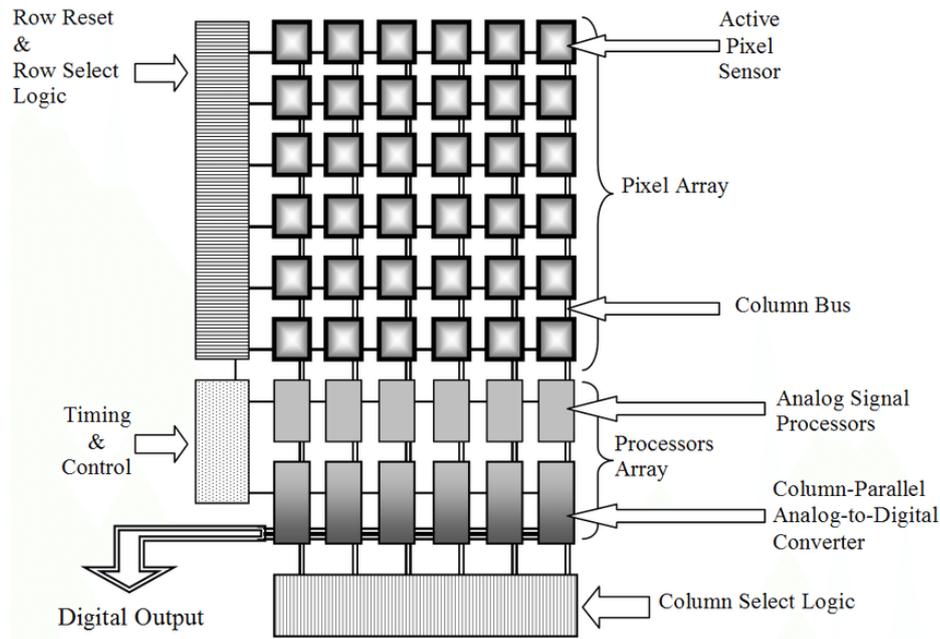


Figure 1.2: CMOS image sensor architecture

can achieve 12 to 13 bits resolution and operate faster. However highly requiring of components linearity makes it hardly achieve more. SAR ADCs provide low power consumption and a relatively high speed compared to other ADCs. However requiring of large area caused by capacitors in DAC decreases the compatibility with the size of pixel pitch. With a segmented capacitor DAC, [9] and [10], of which sizes are adapted to widths of their pixels, and achieve 12-bit resolution. Over-sampling ADC, such as sigma-delta ($\Sigma\Delta$) ADC has been investigated for CMOS image because it requires relatively low precision analog components and reduces input temporal noise, compared to conventional Nyquist-rate converters. Although S/H module is suppressed, $\Sigma\Delta$ ADC works as incremental $\Sigma\Delta$ ($I\Sigma\Delta$) ADC in image sensor application for the reason that modulator is reset before every conversion beginning. Column-parallel $\Sigma\Delta$ ADC in [11] compatible with pixel pitch by using inverter-based amplifier [12] instead of operational amplifier, achieves 12-bit resolution and 220-kS/s conversion rate. It's possible to obtain higher resolution by doubling the conversion time in theory. Also using inverter-based amplifier, [13] has 14-bit resolution and 100-kS/s conversion rate with the width of $70\mu m$. The summary of different type column-parallel ADCs are shown in Table 1.1

Pure one-step ADCs can hardly take care of both speed and resolution. Furthermore, generally high resolution ADC relies on analog components linearity. In order to achieve high resolution without sacrificing conversion time and increasing circuit complexity significantly, multi-stage ADCs are developed. Pipeline ADC, as shown in Fig. 1.3 can be considered as the most typical multi stage ADC. It is relatively easy to achieve high resolution and medium conversion speed [14]

ADC Type	Resolution	Conversion Speed	Size
SS ADC	++	++	++
Cyclic ADC	++	+++	+
SAR ADC	++	+++	+
$\Sigma\Delta$ ADC or $I\Sigma\Delta$ ADC	+++	++	++

Table 1.1: Summary of performances of different ADCs.

[15][16] while it requires an operational amplifier (OTA) for each stage. It results in immense area and difficulty in integration as column-parallel ADC. It's also feasible to use different ADC at each stage, which is hybrid ADC. With a combination of an $I\Sigma\Delta$ ADC and a cyclic ADC, [17] achieves 14 bits operating at 5 MHz and [18] provides 17 bits with 30-kS/s conversion rate. $I\Sigma\Delta$ +SAR is another choice for column-parallel hybrid ADC. In [19], a 16-bit, 12.8-kS/s ADC is developed, whose width is 40 μm . [20] proposed a 14-bit, 150-kS/s ADC with relatively large width. Anyway, multi-

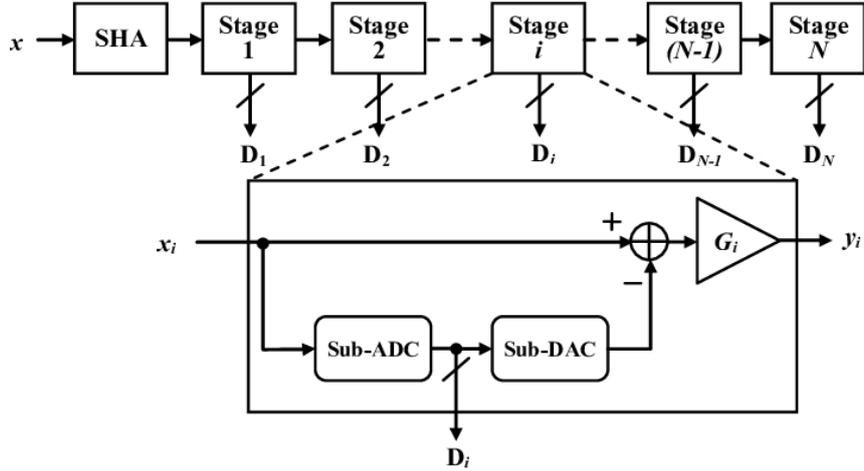


Figure 1.3: Pipeline ADC architecture

stage ADCs require additional circuit resulting in area increasing. On the other hand, multi-step ADCs, whose basic structure is shown in Fig. 1.4, collect the residue(R_i) of each step and estimate the R_i in step $i + 1$ in order to have fine estimation of ADC input. Though conversion over multi steps cannot be operated in pipeline as certain multi-stage ADCs, it simplifies the circuit by reusing materials for all the steps. Two-step single-slope ADC [21][22] can achieve 12 bits with an acceptable conversion time. While Two-step SAR ADCs respectively proposed in [23] and [24] have 14-bit resolution. Although their widths are compatible to the pixel pitch size, their length are much larger than other ADCs. In [25], an $I\Sigma\Delta$ ADC in two-step is developed, obtaining 15 bits and 20-kS/s conversion rate. Converting in two steps, overall number of clock cycles decreases from 2^n to $2^{\frac{n}{2}}$. Despite of its low clock frequency, another two-step $I\Sigma\Delta$ ADC [26], reduces further more the overall number of clock cycles by using a second-order $I\Sigma\Delta$ modulator.

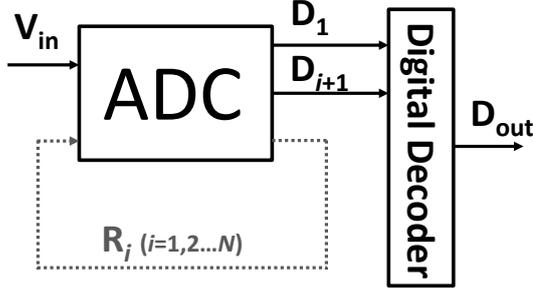


Figure 1.4: Multi-step ADC architecture

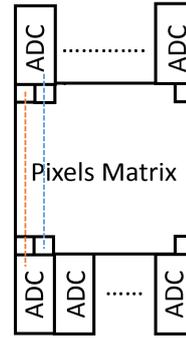


Figure 1.5: Pixel Matrix with two columns of ADCs in parallel.

1.2 Challenge of ADC in the Earth Observation Application

Featuring significant advantages over CCD sensors (lower power consumption, integration monolithic...), CMOS technology is also expanding in some high-end space applications, such as the Earth observation which requires high-resolution, medium-speed as well as high-dynamic-range image sensors. [27] developed a 750×750 pixels CIS specifically designed in order to fit star trackers. Nowadays we use a full HD CIS with 1080×1920 pixels and 100-frame/s data rate to increase resolution and speed. Therefore, the ADC conversion rate should not be lower than 100 kS/s to allow CIS reading out 100 images per seconds. To guarantee high dynamic range, we challenge to design a column-parallel ADC with high resolution of effective 14 bits. To release the constraint of the ADC width, we place half of column-parallel ADCs at the top of pixel array and half of them at the bottom, as the CIS architecture shown in Fig. 1.5. However the width of each ADC still should not exceed the width of two pixels.

1.3 Pieretta: Chip of New Proposed Two-step $\Sigma\Delta$ ADCs

Pieretta[28] is our response to the challenge. It is the name of the chip of two-step $\Sigma\Delta$ column ADCs. This ADC architecture was proposed by a previous PhD student Pierre Bisiaux who defended in 2016.

Acting as a Nyquist rate converter, $\Sigma\Delta$ ADC is an ideal candidate for medium-frequency and high-resolution application. Different from $\Sigma\Delta$ which focus on the output's spectral properties, $\Sigma\Delta$ ADC provides an accurate estimation for every individual sample. What's more, because of its memory-less conversion, it can be easily switched between different signal sources. These characters make it suitable for image sensor applications. For this reason, Pierre decided to

develop the ADC based on $\Sigma\Delta$ ADC.

Conversion rate was an important consideration when the architecture of $\Sigma\Delta$ ADC (modulator) was chosen. The use of 2^{nd} order $\Sigma\Delta$ modulator provides a good trade-off between the number of clock cycles needed for a conversion and the circuit complexity. Once the order of the $\Sigma\Delta$ modulator has been chosen, he found out the modulator input range based on Fig. 1.6 so that the quantization errors stay in a reasonable zone as the effective number of bits depends on the range of quantization errors. Fig. 1.7 shows the quantization errors in the function of the input of the $\Sigma\Delta$ modulator when 25 clock cycles (OSR) are applied. According to it, if the input amplitude is limited to 0.8 reference voltage (the value of full scale voltage equals to two times the value of reference voltage), the quantization errors cover to a normal range. This conclusion is also valid for $OSR = 50$ and $OSR = 100$, as shown in Fig. 1.8.

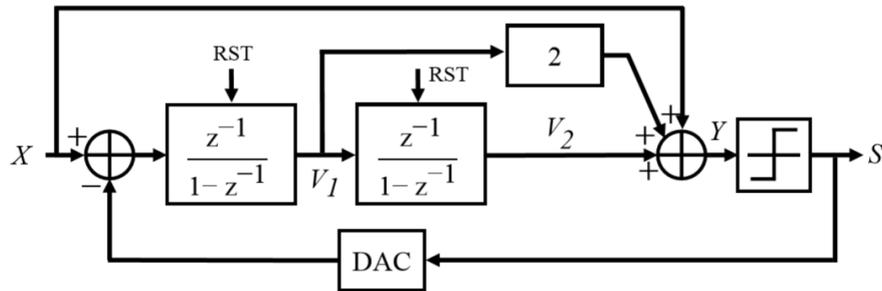


Figure 1.6: Architecture of a second-order $\Sigma\Delta$ modulator.

By choosing a two-step conversion approach, we can further reduce the number of clock cycles without extra circuit. Mr. Bisiaux also optimized the coefficients of the integrators in the $\Sigma\Delta$ modulator for two purposes: first, making the range of residue in of first step close to the range of the modulator input, in order to eliminate inter-step gain, second, minimizing the residue of the second step so that to minimize the conversion errors. As shown in Fig. 1.9, the distribution of the residue of the first step is more "uniform" than that of the residue of the second step. Thus for our architecture, if the ADC operates at a 20 MHz system clock with 72 clock periods, it is possible to achieve a conversion rate of 250 KS/s with a resolution of 14 bits, which provides the same performance as a one-step ADC with 286 clock periods.

As a column ADC, in order to fit its width with the pixel pitch, inverter-based amplifiers are employed instead of operational amplifiers (OTAs). CMOS inverters have also been found in the literature as being used for realizing OTAs for $\Sigma\Delta$ ADCs, because of its simple structure and [12][29][30]. For low power applications, to reduce the power consumption of inverter-based OTAs, they are operated at lower supply voltages [31][32]. It is namely class-C operation where the supply

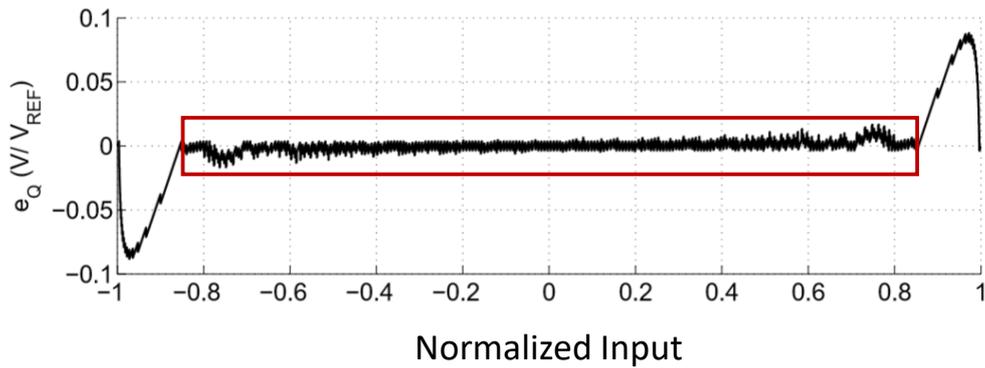


Figure 1.7: Quantization errors of the $\Sigma\Delta$ modulator in the function of the input amplitude when 25 clock cycles are applied.

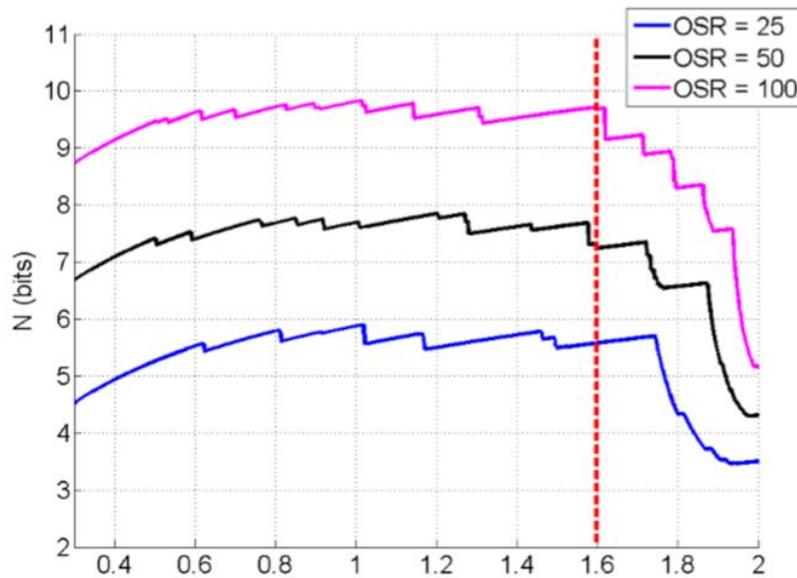


Figure 1.8: Equivalent number of bits of the $\Sigma\Delta$ modulator in the function of the input amplitude for some values of OSR .

voltage is chosen slightly lower than the sum of the threshold voltages of NMOS and PMOS transistors ($|V_{thn}| + |V_{thp}| > V_{DD}$). However, in our application, power consumption isn't critical for design. To the opposite, a strong current is required in order to ensure a sufficient GBW and slew rate which are not guaranteed by class-C OTA. In order to use a minimum area to deliver this high current, a class-AB amplifier is therefore used. For this type of inverter, it is easy to obtain a GBW of 100 MHz, a value necessary in order to satisfy the time constraints of the chosen architecture. Since these OTAs are used for implementing integrators, significantly high dc gain is desirable, in order to minimizing the error of the integrators. To increase the DC gain of amplifier,

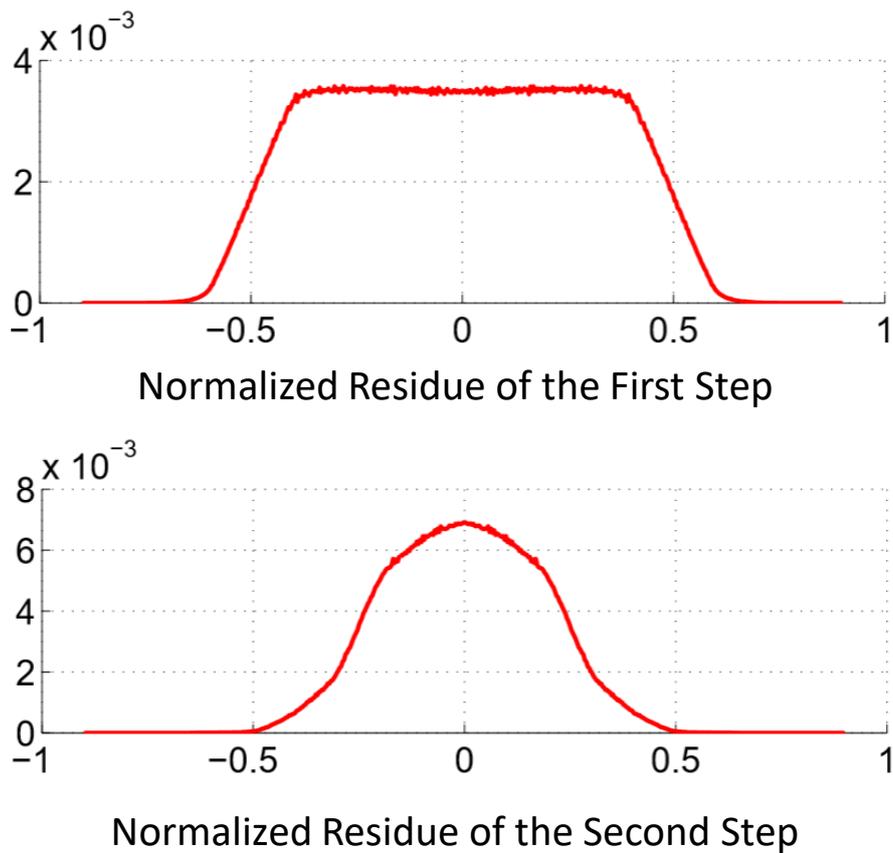


Figure 1.9: Relatively occurrence frequency of the residues of each steps when 200 clock cycles are applied.

cascading and gain-boosting techniques, as shown in Fig. 1.10, have been applied [33][34][35]. These techniques increases the output impedance and thereby increases the DC gain of the amplifiers. To ensure 14 bits ADC resolution, a DC gain of 80 dB of amplifiers is sufficient. Finally, Mr.Bisiaux employed cascading and gain-boosting techniques for designing the amplifiers in our circuit. Table 1.2 shows the summary of the inverter-based amplifiers designed by Mr.Bisiaux.

Pierre also finished the design of analog circuit as well as a testing digital circuit of the $I\Sigma\Delta$ modulator and validated his design by post-layout simulation. The summary of the $I\Sigma\Delta$ modulator is shown in Table 1.3.

1.4 Motivation and Problematic

The OTA non-idealities are responsible for incomplete or inaccurate transfer of charge between the capacitors in integrators [36], that is so call a leakage in the integrator [37]. Orla Feely and Leon O Chua study the effect of integrator leak in a first-order single-loop $\Sigma\Delta$ modulator in [37]

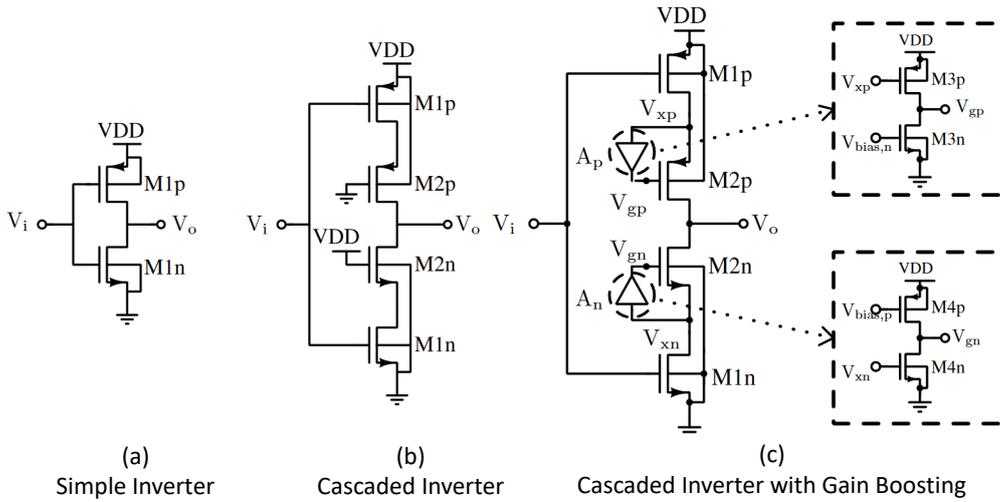


Figure 1.10: Diagrams of inverter-based amplifiers: (a) Simple inverter; (b) Cascaded inverter; (c) Cascaded inverter with gain boosting.

Class	AB
Power supply	1.8 V
Biasing Current	76 μA
Load capacitance C_L	400 fF
DC gain	80 dB
linear input range	0.9 V
linear input range	0.9 V
GBW	386 MHz V
Power consumption	136 μW

Table 1.2: Summary of inverter-based amplifiers designed by Pierre.

by looking at the dynamic range of the integrator values in each cycle. With an ideal integrator, applying a DC input, the modulator outputs are periodic (with a number of limit cycles) and its average equal to the input[38]. And with the integrator leak, the modulator outputs may still be periodic, however the number of limit cycles varies in the function of the modulator inputs. As a results, using a fixed number of limit cycles to calculate the average modulator output, we may obtain a value different to the modulator input.

Finite DC gain (DCG) is one of the source leading to integrator leakage. Many researches about its impacts on SC integrators and $\Sigma\Delta$ modulators have been done by modeling the transfer function of SC integrator in z domain[39][40][41][42]. Finite DC gain, in z-domain, leads to the shift of poles and zeros in the transfer function of the integrators. Further more, if the gain is non-linear (output signal-depending, as shown in Fig. 1.11), the coefficients of the poles and zeros are in

States	Descriptions
Technology	180 nm
Power supply	1.8 V
Sampling Frequency	250 kHz
Power Consumption	$460\mu W$
Surface	$25\mu m \times 370\mu m$
SNDR	72.8 dB

Table 1.3: Summary of the $\Sigma\Delta$ modulator.

the transfer function are not constant. Charge transfer error (leakage) of integrators finally results in distortion of the $\Sigma\Delta$ modulator outputs in frequency domain. [43] and [42] study the effects of finite and nonlinear DCG of the Amplifiers in $\Sigma\Delta$ Modulators employing SC integrators by defining two indicators: Finite DC-Gain (FDCG) which refers to the maximal DC gain and DC-Gain (DCG) Linearity indicating the degree of amplifier gain linearity (A_0 in Fig. 1.11). The simulation results (in Fig. 1.12) of a second-order $\Sigma\Delta$ modulator on SIMULINK MATLAB shows that: to ensure a SNDR of 90 dB, (a) if the amplifier gain is very non-linear (DCG linearity < 10 dB), a very high FDCG is required (above 10000). (b) There is a linear-gain part of the curve in (in Fig. 1.12) where the required DCG linearity increases by 20dB/oct of the required FDCG. (c) if DCG linearity is above 50dB, the FDCG of 100 is enough. [42] indicates that non-linear finite OTA gain introduces odd harmonic distortion to the output of the modulator and moreover increases the in-band noise level. As shown in Fig. 1.13, for the same $\Sigma\Delta$ modulator in [43], when DC gain (DCG) linearity drops from a moderate value (37dB) to very low value (9dB), SNDR decreases from 96 dB to 69 dB. Unfortunately, one of the well-known issue of inverter-based amplifiers is its non-linear finite amplifier gain. [39] points out that the finite amplifier gain may even endanger the modulator stability.

Other OTA non-idealities such as finite GBW (gain-bandwidth product) and SR (slew rate) of OTA may degrade the performance of SC integrators. WJ Wolski in [44] points out that SR of the op-amp (operational amplifier) and time constants caused by non-zero switch-on resistance are main effects which limit the maximal frequency of sampling in SC network. The finite gain-bandwidth product and switch-on resistance introduce the parasitic poles and zeros in the transfer function of integrator, thus producing error of amplitude and phase frequency characteristics. However, when sampling frequency is small, the finite gain-bandwidth product and switch-on resistance effect only the integrator gain as the parasitic poles and zeros lie very near by the z-plane origin. For $\Sigma\Delta$ modulators, the SC integrator non-imperfection caused by the finite GBW and SR of OTA produce dynamic errors and harmonic distortion, thus degrading the SNDR performance

of the $\Sigma\Delta$ modulators [40][36].

Apart from it, different to differential inverter-based OTAs in [45][46][47][48], our amplifiers have only single input. Thus the common mode voltage V_{cm} of input makes difference to the amplifier characters as it cannot be eliminated. Additionally, for a high resolution ADC, any other defects (causing linear or non-linear errors) such as capacitor mismatch [49][50] and charge injection [51][52] in switches may also hinder it from achieving the desired accuracy.

We can obtain a conclusion of the previous small research of ADC circuit non-idealities and their impacts: just designing ADC is not the end of our work for a goal of high resolution. Some techniques should be considered to compensate the degradation brought by these circuit non-idealities. We call these techniques as calibration techniques. The objective of my research is to develop a calibration technique for our two-step $\Sigma\Delta$ ADC so that it can achieve the target resolution regardless of circuit imperfections.

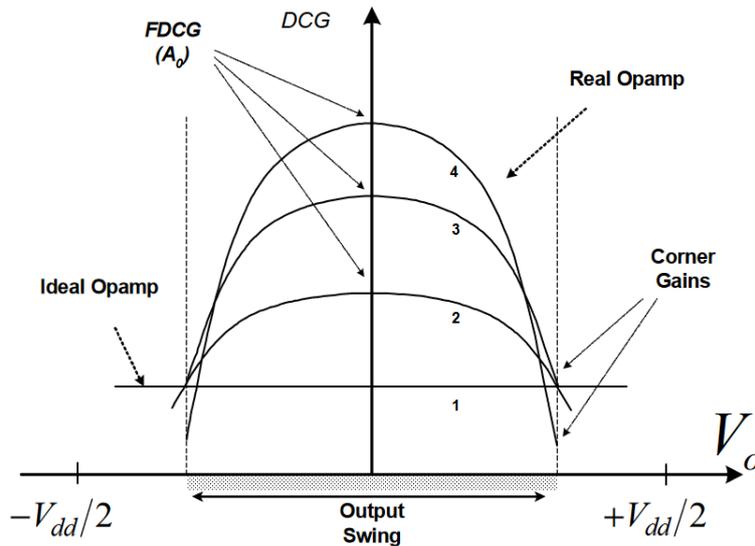


Figure 1.11: A typical op-amp's DCG versus output voltage with the rail-to-rail voltage of V_{dd} .

1.5 Thesis Organization

This dissertation is organized as follows:

Chapter 2 first narrates evolution of incremental sigma-delta analog-to-digital converter ($I\Sigma\Delta$ ADC). Then it introduces conventional calibration techniques for $\Sigma\Delta$ ADCs and optimization techniques for $I\Sigma\Delta$ ADCs and describes the limitation of each existing technique.

Chapter 3 introduces circuit Pieretta in terms of mathematical modeling and circuit design. The mathematical model shows that our $I\Sigma\Delta$ ADC performs over two steps has great advantage

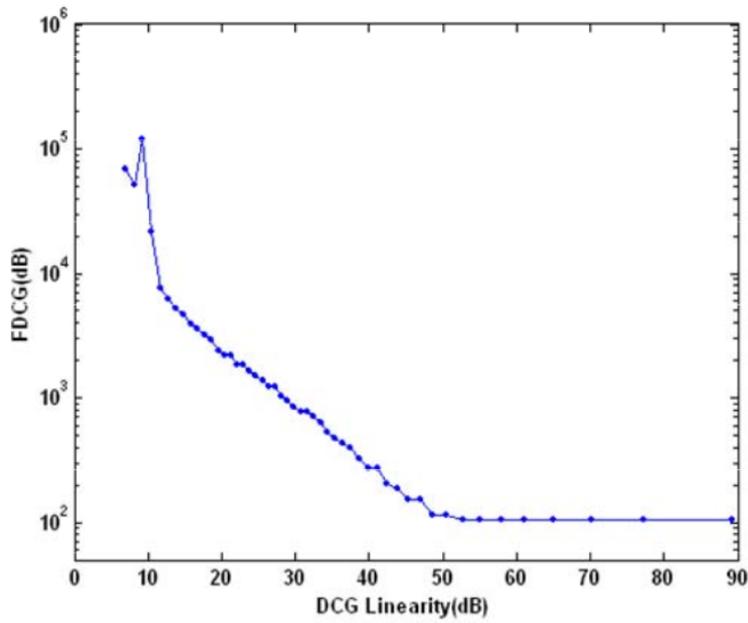


Figure 1.12: The FDCG and DCG linearity required for the output SNDR nearly 90dB.

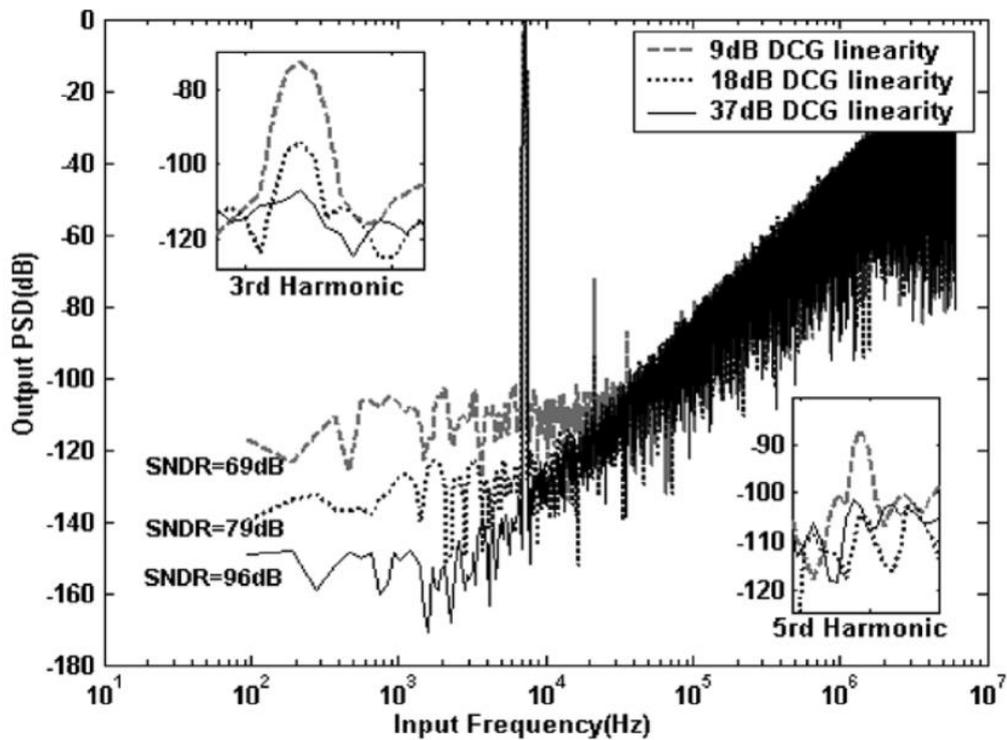


Figure 1.13: The PSDs of the modulator output with the first integrator nonlinear DCG of 9, 18, and 37 dB.

in conversion rate compared to classical one-step $\Sigma\Delta$ ADCs. Apart from it, reuse of circuit during two-step conversion avoid auxiliary components that saves circuit area. Circuit design includes

two parts: the analog part shows schematics and dimensions of transistors of each component while the digital part illustrates control flow of the digital circuit and interfaces of our circuit.

the proposes the output-referred distortion analysis of the $\Sigma\Delta$ modulator, which takes into account the error circulation in the DTI. From this analysis, the universal model including multiple nonlinear error sources is proposed. The similarity with the exiting method is provided to give more insight into the proposed approach. The mismatch effect of PN injection in modeling is also considered in this chapter.

Chapter 4 first presents classical correction methods for $I\Sigma\Delta$ ADCs. However according to post-layout simulation, these classical correction methods are not completely suitable for our circuit. Therefore, it then proposes a new correction methods considering the ADC as a black-box. According to post-layout simulation, it is able to improve resolution significantly. Next, it simplified this new proposed correction method to make it implementable. At the end, our ADC achieve the target resolution after the correction with an acceptable complexity.

Chapter 5 proposes an error model of inverter-based integrators which are employed in our $I\Sigma\Delta$ ADC. This error model can well explicate the behavior of our circuit in post-layout simulation as well as most part of the degradation. With the help of the error model, we can further study characters of our ADC and correction methods.

Chapter 6 presents test-bench, sampling and filtering method of the ADC as well as measurement results. Regardless of a global shift in resolution for all correction methods in measurement, the tendency is coherent to that in post-layout simulation.

Chapter 7 concludes and summarizes this research and discuss the future work as well.

Chapter 2

State of the art

The core of $\Sigma\Delta$ ADCs or $I\Sigma\Delta$ ADCs are $\Sigma\Delta$ modulators. After decades of development, there are diverse architecture of $\Sigma\Delta$ modulators. The order of $\Sigma\Delta$ modulators have been increased from first order at the beginning to until fourth order so as to obtain higher resolution during the same clock cycles. In order to enhance the stability of $\Sigma\Delta$ modulators, multi-stage architectures appeared. Apart from pure $\Sigma\Delta$ ADCs, there are also some ADCs combined $\Sigma\Delta$ modulator with other ADC architectures also for the purpose of increasing resolution without sacrificing conversion speed.

In reality, sometimes we cannot get the resolution as we expected because of circuit imperfections, especially for the high-order ADCs. Therefore, a calibration is required. Some calibration methods for $\Sigma\Delta$ ADCs already exist, however, they are only suitable for those with multi-bit quantizer. Even if the architectures of $I\Sigma\Delta$ ADCs are similar to those of $\Sigma\Delta$ ADCs, there are few calibration methods for $I\Sigma\Delta$ ADCs with circuit imperfections. Most of the researches about $I\Sigma\Delta$ ADCs focus on finding a better way to reconstruct the inputs of ideal $I\Sigma\Delta$ ADCs.

2.1 Incremental $\Sigma\Delta$ ADC

The principle of incremental $\Sigma\Delta(I\Sigma\Delta)$ ADCs is $\Sigma\Delta$ modulation which is a method of pulse-code modulation (PCM) encoding. [38] proposes to apply PCM in an analog-to-digital converter, the architecture of which is shown in Fig. 2.1. It uses a negative feedback loop including an integrator and a quantizer. The digital outputs of the quantizer are accumulated during certain cycles and an output is taken from the accumulator. Then its content sets to zero before cycling for the next sample. An A/D converter of basic $\Sigma\Delta$ modulator is implemented in the form of a bipolar integrated circuit [53]. Its schematic is shown in Fig. 2.2. Implemented with CMOS, [54] applies

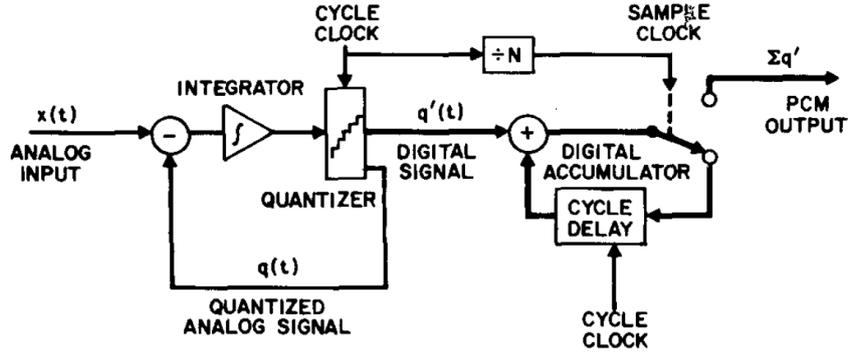


Figure 2.1: A/D converter based on PCM

reset signal not only in the digital counter but also in the $\Sigma\Delta$ modulator, as shown in Fig. 2.3. It can be considered as the origin of $\Sigma\Delta$ ADCs. With $V_{in} \in [0, V_R]$, at the beginning of a new conversion, the integrator and the counter are both reset. Then, a fixed number ($N = 2^{nbit}$) of discrete integration steps are performed, where $nbit$ is the target resolution in bits. Whenever the input of the comparator exceeds zero, its output becomes 1, and V_R added to the input of the analog integrator. After $N = 2^{nbit}$ clock cycles, the next output of the delaying integrator (which is bounded by $(-V_{ref}, V_{in})$) would become:

$$V_{out}[N + 1] = NV_{in} - N_{out}V_R \quad (2.1)$$

where N_{out} is the number of clock cycles when feedback was applied. Since the known reference voltage V_R is applied to the integrator input only when the output of the comparator is 1, it is called unipolar. As $V_{out}[i]$ always satisfies $-V_R < V_{out}[i] \leq V_{in} \leq V_R$, we have:

$$V_{in} = \frac{N_{out}V_R}{N} + \epsilon \quad (2.2)$$

where $\epsilon \in [-\frac{V_R}{N}, \frac{V_R}{N}]$. Thus, $\frac{N_{out}V_R}{N}$ is a digital representation of V_{in} , with a quantization error ϵ .

The incremental converter can also be implemented in bipolar ways by enabling bipolar input signal ($V_{in} \in [-V_R, V_R]$) and feeding back bipolar reference signal instead of unipolar. In this way, the digital representation of V_{in} becomes $\frac{V_R \sum_{i=1}^N s(i)}{N}$, where $s(i)$ is the output of comparator.

Over 40 years, the architectures of $\Sigma\Delta$ modulator have been developed and improved several way, because of the fundamental drawback of the first-order $\Sigma\Delta$ A/D converter: it must be operated through 2^{nbit} clock cycles to achieve $nbit$ -bit resolution. Therefore, the conversion (output) rate is extremely slow compared to the circuit's clock frequency. To solve this problem, there are

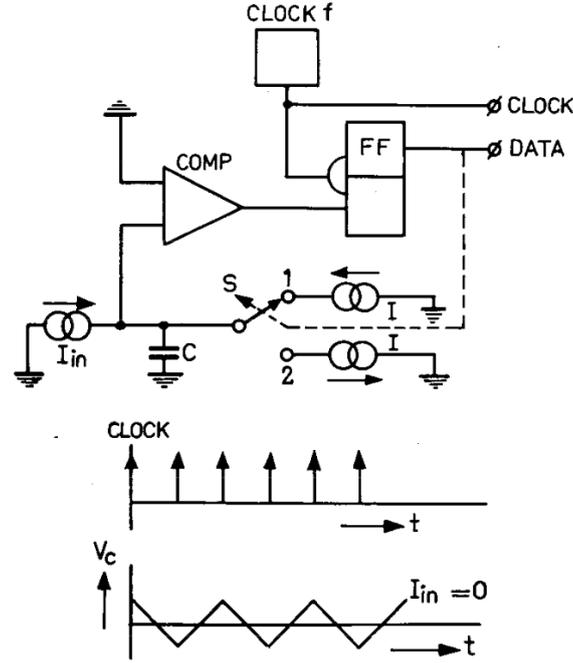


Figure 2.2: A basic sigma-delta A/D converter implemented with bipolar

two ideas.

One ideal searches for different modulator structure which leads to the raise of second or high-order $\Sigma\Delta$ modulator structure [55][56][57]. An example of the architecture of a third-order $I\Sigma\Delta$ ADC in [58] is shown in Fig. 2.4.

After a M clock cycles conversion, the ADC input V_{in} can be presented as:

$$d = \frac{V_R \cdot 3!}{(M-2)(M-1)M} \sum_{k=0}^{M-1} \sum_{j=0}^{k-1} \sum_{i=1}^{j-1} s(i) \quad (2.3)$$

And the equivalent value of the LSB voltage can be found as:

$$V_{LSB} = \frac{2 \cdot 3! V_R}{(M-2)(M-1)M} \cdot \frac{1}{c_1 c_2} \quad (2.4)$$

Thus the equivalent number of bits (ENOB) can be estimated by:

$$\begin{aligned} nbit &= \log_2 \left(\frac{2V_R}{V_{LSB}} \right) \\ &= \log_2 \left(\frac{c_1 c_2 (M-2)(M-1)M}{3!} \right) \\ &\approx 3 \log_2(M) + \log_2(c_1 c_2) - 2.6 \end{aligned} \quad (2.5)$$

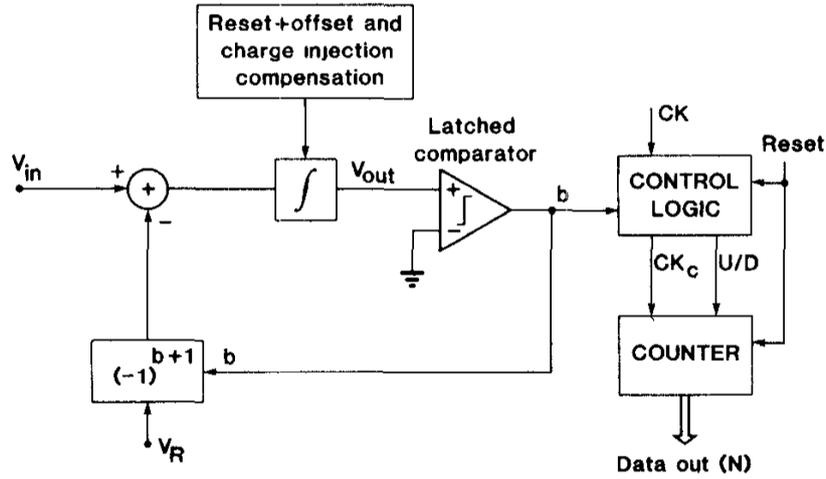


Figure 2.3: Uni-polar first-order incremental sigma-delta A/D converter

For single-loop $\Sigma\Delta$ modulators, we can even see an architecture of fourth order [59]. Another extension structure of $\Sigma\Delta$ modulator is that described in [60]: multi-stage noise shaping (MASH), which consists of two first-order $\Sigma\Delta$ loops, as shown in Fig. 2.5. In this architecture, the digital representation of the ADC input, d can be expressed as:

$$d = \frac{V_R}{2} \left(\sum_{i=1}^p a_1(i) \cdot (p+1-i) + \sum_{i=2}^{p+1} b_2(i) \right) \quad (2.6)$$

where p is the number of clock cycles in one conversion. In this case, $V_{LSB} = 4V_R/p(p+1)$ and ENOB can be estimated by:

$$\begin{aligned} nbit &= \log_2 \left(\frac{2V_R}{V_{LSB}} \right) \\ &\approx 2\log_2(p) - 1 \end{aligned} \quad (2.7)$$

For 16-bit accuracy, N is reduced to 362 instead of 2^{16} . In addition, by detecting the sign of the output of the integrator in the second stage at the end of the conversion, an extra bit of resolution can be obtained. In this way, N can be further reduced to 257.

Thanks to development of $\Sigma\Delta$ modulator structures many high resolution $\Sigma\Delta$ ADCs have been realized in the last 20 years [61, 62, 63, 64, 65]. The other solution to reduce N is to refine

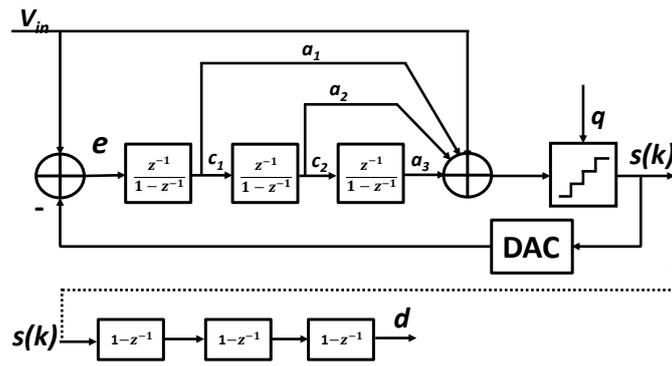


Figure 2.4: High-order IΣΔ ADC architecture

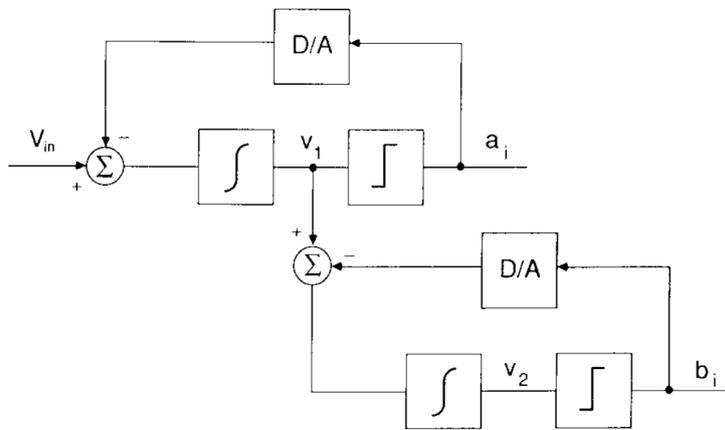


Figure 2.5: MASH 1-1 IΣΔ ADC architecture

the quantization error.

[66] proposes a two-step pipeline IΣΔ ADC (as shown in Fig. 2.6) which doubles the second-order IΣΔ modulators. The first stage IΣΔ modulator gives a coarse estimation of the ADC input while the second stage allows refining the quantization error by estimating the residue of the first step. In this case, global numbers accuracy equal to the sum of accuracy of each IΣΔ modulator.

On the other hand, also for the purpose of increasing conversion speed, many attempts have been made to combine IΣΔ ADCs with other ADCs. [67] is composed of an 11-bit SAR and a 4-bit IΣΔ ADCs for the coarse and fine conversions, and its architecture is shown in Fig. 2.7.

At the coarse SAR conversion, the circuits VCDL in Fig. 2.7 play the role of an open-loop pre-amplifier. And VCDL and the quantizer operates as a comparator that decides the polarity for the 11-bit SAR ADC conversions with the INLS switching procedure which decides the bottom plate potential of SAR capacitor array. After the SAR conversion, the residue voltage remains on the top plates of the capacitor arrays and it is maintained along during the fine IΣΔ conversion. In the IΣΔ conversion, VCDL acts as an oscillator. To be noticed that these two different ADCs

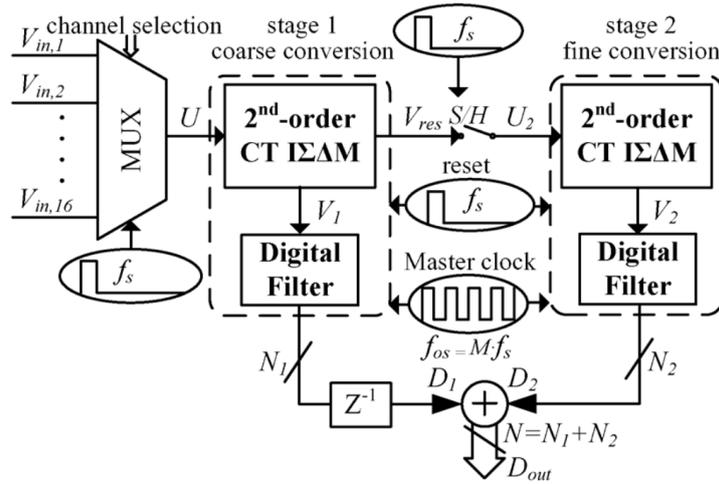


Figure 2.6: Architecture of two-step pipeline $\Sigma\Delta$ ADC

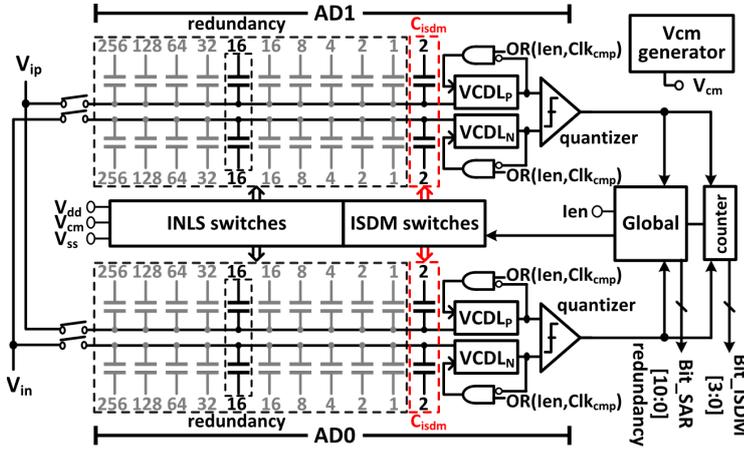


Figure 2.7: Hybrid SAR- $\Sigma\Delta$ ADC architecture

share the same operational amplifiers to save circuit area. The share-amplifier idea is applied in [17][68], as shown in Fig. 2.8 which combines $\Sigma\Delta$ ADC and cycle ADC together.

2.2 Linearization Techniques for $\Sigma\Delta$ ADCs

Since both $\Sigma\Delta$ ADCs and $\Sigma\Delta$ ADCs employ a $\Sigma\Delta$ modulator, the linearization and calibration methods for $\Sigma\Delta$ ADCs may provide inspiration to us. Compared to linear error, non-linear error is more difficult to calibrate for $\Sigma\Delta$ ADCs. The linearization techniques $\Sigma\Delta$ ADCs are usually developed for two main functional blocks, the integrators or the feedback DACs [69][70]. Since our circuit employs a one-bit DAC, we only focus on the linearization techniques for the integrators. One idea of the linearization techniques is to enhance the amplifier linearity in the integrators

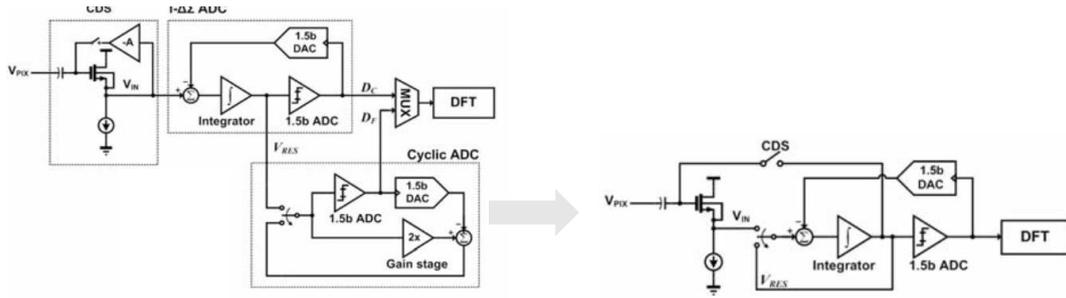
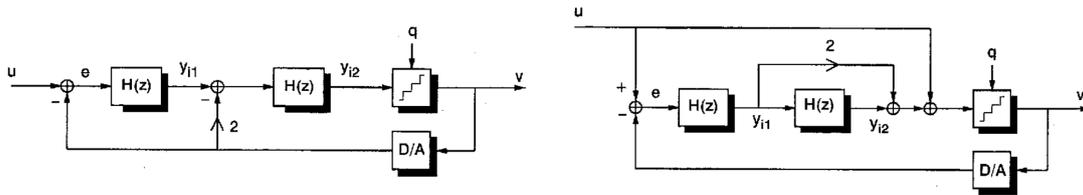


Figure 2.8: Architecture of an Extended Counting $I\Sigma\Delta$ ADC in $I\Sigma\Delta$ conversion



(a) Traditional topology.

(b) Reduced distortion topology.

Figure 2.9: Topology of a second-order $\Sigma\Delta$ modulator

by applying low distortion structures in [71][72][73]. Fig. 2.9a[74] shows an example of a low distortion $\Sigma\Delta$ modulator topology comparing to a traditional one shown in Fig. 2.9b. A direct feed-forward path from the analogue input to the quantizer is added, which ideally enables the perfect cancellation of the input signal at the input of the loop filter. Therefore the integrator does not process the input signal but only handles the quantization noise shaping. However, it is essentially an optimization in design step rather than a calibration technique.

The other idea of the linearization techniques is to find the calibration techniques for different $\Sigma\Delta$ ADC structures. For a calibration method, generally there are three parts: error modeling, cancellation, and parameter identification.

[75] proposes a classical model of a non-ideal integrator which is one of the main functional blocks determining the linearity in $\Sigma\Delta$ ADCs as it is well known that the amplifier gain is finite and non-linear. For weakly nonlinear amplifiers, the non-linear amplifier gain $A(n)$ can be well approximated by an N th-order power series of current amplifier output $y(n)$:

$$\frac{y(n)}{A(n)} \approx \frac{\sum_{i=1}^{N-1} k_i y^i(n)}{A_{max}} \quad (2.8)$$

where A_{max} is the maximum amplifier gain and $k_1 = 1$ while the higher-order coefficients are determined by the amplifier character as shown in Fig. 2.10. This is equivalent to stating that the amplifier gain is time-varying depending on its output Y .

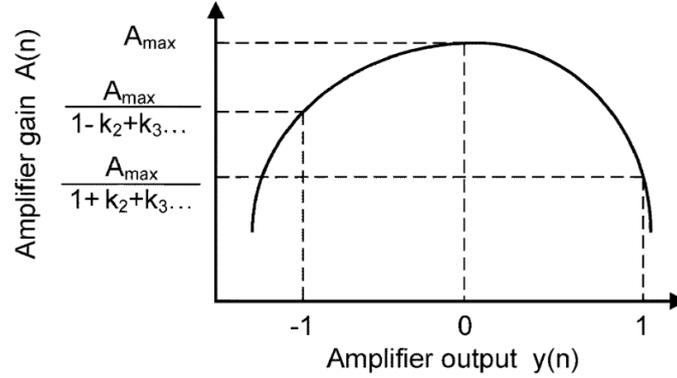


Figure 2.10: Polynomial model of the output-signal-dependent amplifier gain

A typical processing of the finite non-linear amplifier gain error leads to the pole and zero movements embodied in a z-domain model of an integrator (whose architecture is shown in Fig. 2.11) given by:

$$Y(z) = \beta(A)z^{-1}Y(z) + \alpha(A)z^{-1}W(z) \quad (2.9)$$

where W and Y are respectively integrator input and output while $\beta(A)$ and $\alpha(A)$ are functions of the amplifier gain. According to equation (2.8), the value of A depends on the current amplifier output, as a consequence, $\beta(A)$ and $\alpha(A)$ are also signal-dependent functions. That means that their values are not constant because of the varying amplifier output in each cycle.

Based on a non-ideal integrator model, [75] proposes calibration techniques for the first-order or second-order $\Sigma\Delta$ ADCs. However, despite of different order $\Sigma\Delta$ ADCs, the principles of calibration method are the same.

The equivalent model of a non-ideal first-order $\Sigma\Delta$ ADC with time-varying α and β , as shown in Fig. 2.11, can be treated as an ideal modulator plus a time-varying error term E_a which is function of the outputs of each integrator. The equivalent model of a non-ideal second-order $\Sigma\Delta$ ADC is shown in Fig. 2.12. The objective is to compensate E_a ($E_{a1} + E_{a2}$ for second order) with E_d (the estimation of E_a) in order to have a corrected ADC output D_c . The schematic of the ADC with calibration part is shown in Fig. 2.13 (a). Since the integrator outputs are analog values and inaccessible, we usually use the original ADC output D to calculate E_d . In each cycle, the equivalent error $e_d(n)$ is :

$$e_d(n) = -\sum_{i=1}^N a_i d^i(n) + \sum_{i=1}^N b_i d^i(n-1) \quad (2.10)$$

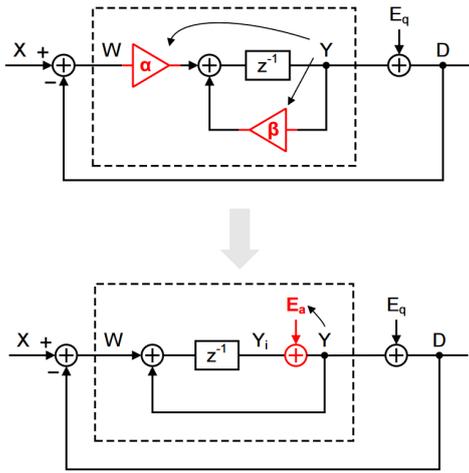


Figure 2.11: Equivalent model of a non-ideal first-order modulator

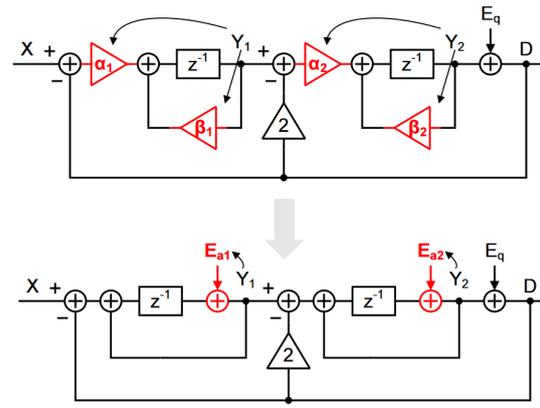


Figure 2.12: Equivalent model of a non-ideal second-order modulator

where $d(n)$ is the ADC output in cycle n and a_i, b_i are functions of A_{max} and k_i . Equation (2.10) is implemented in H_a (as shown in Fig. 2.13 (b)). As a consequence, E_d is only an approximation of E_a and the precision is decided by the precision of the quantizer. For a multi-bit quantizer, we consider E_d is close to E_a . Now we turn the question into how to find the constant coefficients a_i and b_i .

Therefore we need a parameter identification block as shown in Fig. 2.13 (c) to estimate the coefficients k_i in function H_a . The criteria to choose coefficient values is least-mean-square (LMS) of error between D_c and X . Usually a gradient-descent algorithm is employed for parameter learning process. The coefficients finally converge to some stable values when the correlator output is null. A pseudo-random test signal T is injected somewhere in the circuit and removed before D_c . As E_a is the function of integrator outputs, as well as the function of integrator inputs therefore, with test signal, now E_a depends on both ADC input X and test signal T . When E_a is completely compensated by E_d , D_c can represent X , which means correlation between T and D_c is zero as T is pseudo-random.

This calibration can adapt to a MASH $\Sigma\Delta$ ADC in [76] for compensating the noise leakage due to the transfer function mismatch between two stages. The architecture of the MASH $\Sigma\Delta$ ADC is shown in Fig. 2.14. Also utilizing the outputs of quantizers Y_1 and Y_2 , the integrator outputs Y_{ik} ($k = 1, 2, 3, 4$) can be approximated as Y_{di} ($i = 1, 2, 3, 4$) as shown in Fig. 2.15 (a). With Y_{di} ($i = 1, 2, 3, 4$), we can further estimate E_{fdi} ($i = 1, 2, 3, 4$), as shown in Fig. 2.15 (c), which is the approximation of E_i . Fig. 2.15 (b) shows the diagram of correction of the ADC output Y where Y_c is the corrected ADC output. The parameter identification block of a_{i1} and a_{i2} using LMS algorithm

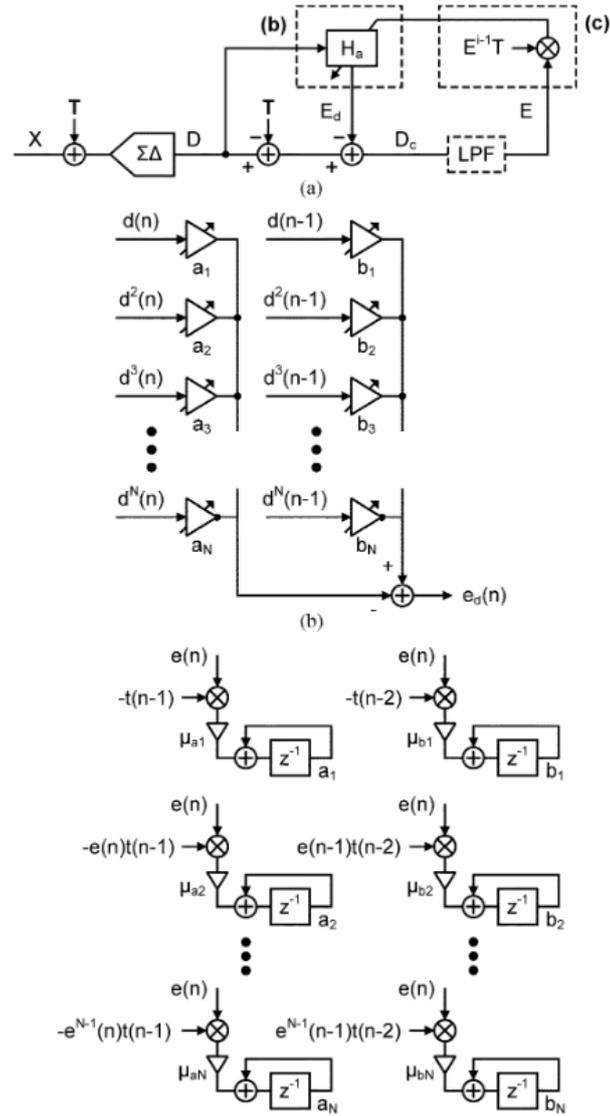


Figure 2.13: Proposed adaptive digital calibration schematic of a first-order $\Sigma\Delta$ modulator: (a) Architecture, (b) error filter (H_a), and (c) LMS update block

is shown in Fig. 2.15 (d).

[77][78] develop a calibration method for a 2-0 MASH $\Sigma\Delta$ ADC of which structure is shown in Fig. 2.16 by applying FIR L_c (as shown in Fig. 2.17 (a)) to compensate the leaked noise. A test signal is also employed for the parameter identification performed by the correlator whose scheme is shown in Fig. 2.17 (b).

However, the digital approximation algorithm of the integrator outputs introduced in [75] and [76] is no longer valid for the $\Sigma\Delta$ modulator where a 1-bit quantizer is employed (our circuit). The same problem occurs in the calibration technique in [77] and [78] which compensates analog circuit imperfections for cascaded $\Sigma\Delta$ ADCs.

There are also other calibration techniques which do not require neither a precise error model

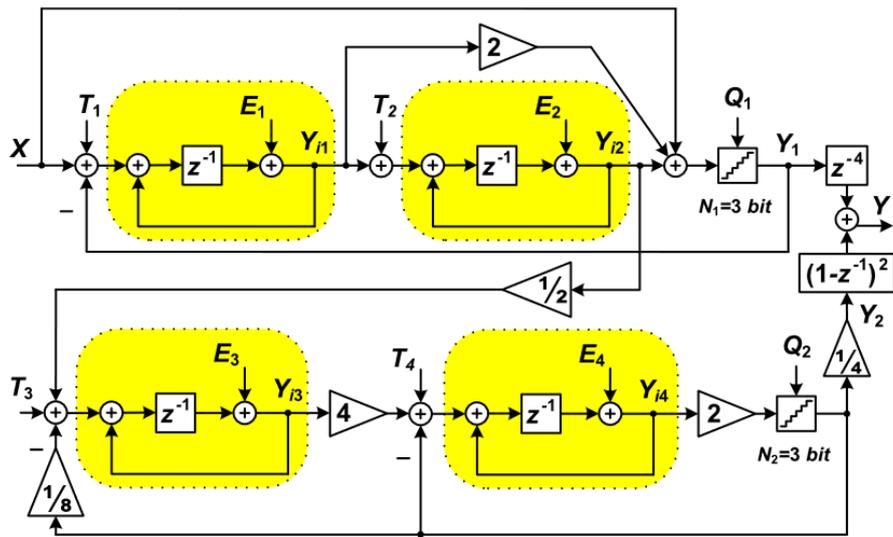


Figure 2.14: MASH 2 – 2 $\Sigma\Delta$ modulator.

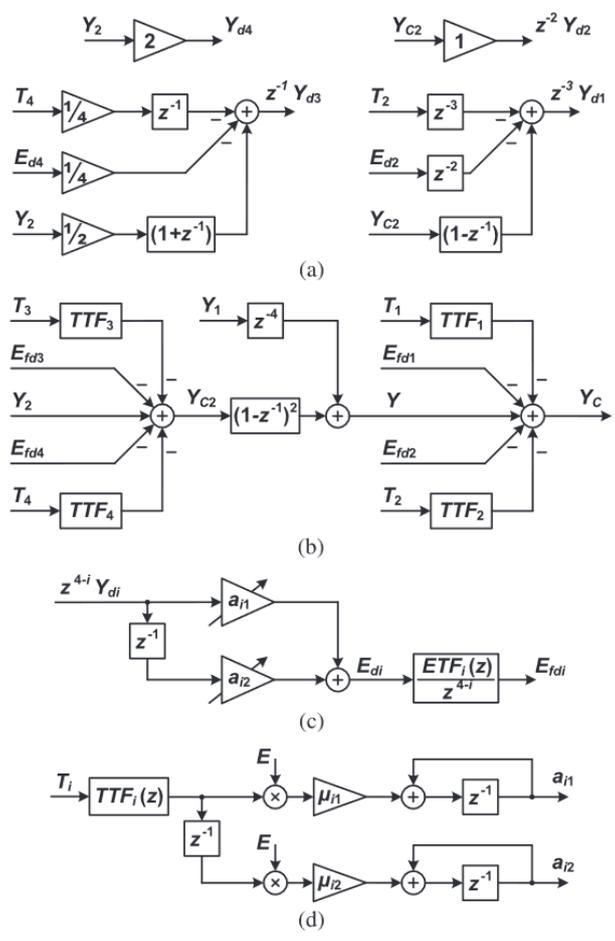


Figure 2.15: Block diagram of (a) integrator output approximation, (b) modulator correction, (c) error estimation, and (d) error parameter identification.

ADC structure.

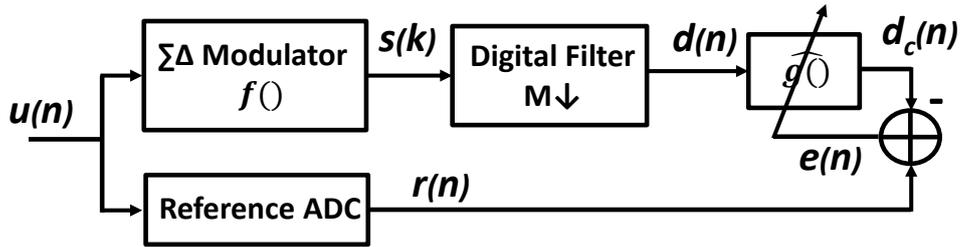


Figure 2.18: ADC in calibration mode: parallel LMS coefficient estimation

2.3 Researches on IΣΔ ADC Reconstruction Filter

For an IΣΔ ADC, the correction and calibration are usually implemented in the digital filter. Even though IΣΔ ADCs and ΣΔ ADCs can share this non-ideal integrator model, so far there isn't any calibration technique dedicated to IΣΔADC while there are several researches on the IΣΔ ADC digital reconstruction filters for the purpose of resolution improvement.

As an example in Fig. 2.19, applying a digital reconstruction filter whose order is higher than the order of analog modulator allows increasing resolution and the average accuracy, but the range of quantization error around zero remains the same [81]. As shown in Fig. 2.20 (b), compared to Fig. 2.20 (a), quantization errors applying higher order digital filter are more smooth except the peak around zero input. To eliminate this peak, we can inject a dither signal into the loop before quantizer as it prevents the oscillation of the integrator output. The quantization errors with dither signal are shown in Fig. 2.20 (c). Since it can be considered as a high frequency quantization noise, it will be filtered by modulator.

A sinc filter can provide the suppression of a periodic disturbing signal and it is advised to chose its order L as $L_a + 1$, where L_a is the order of the analog modulator [81].

However they are also only valid in ideal circuit.

For IΣΔ ADCs, [82] proposes a digital decimation filter (whose transfer function in z domain is $H(z)$) which is able to minimize the sum of modulator input thermal noise and quantization error. By knowing modulator STF and NTF with finite-length impulse response, this digital filter can be mathematically derived. The noise model of an IΣΔ ADC can be abstracted as the model shown in Fig. 2.21. Since the incremental ADC is reset every M clock period, the final result of the n th

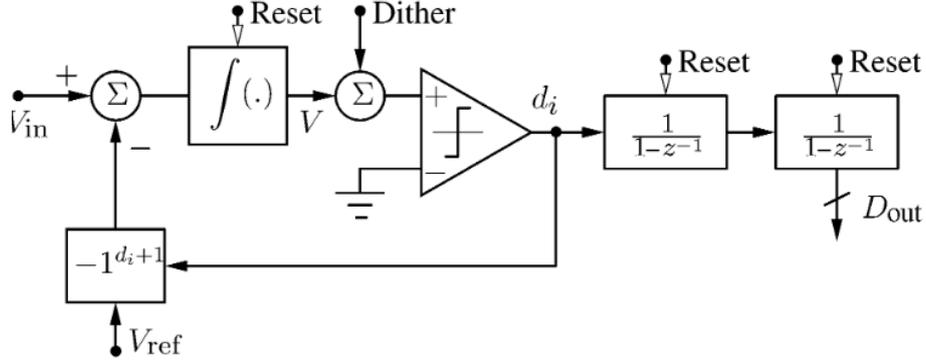


Figure 2.19: Improved first-order converter with second-order filtering and dither.

conversion cycle can be calculated as follows:

$$v(n) = [stf'(k) * u(k)]_{M,n} + [stf'(k) * t(k)]_{M,n} + [ntf'(k) * q(k)]_{M,n} \quad (2.11)$$

where $stf'(k)$ is the M sample impulse response of the overall signal transfer function $STF(z)H(z)$, and $ntf'(k)$ is the M sample impulse response of the overall noise transfer function $NTF(z)H(z)$. Supposing that $t(k)$ and $q(k)$ are both signal-independent, the M -length impulse response of decimation filter $H(z)$ allowing minimizing $[stf'(k) * t(k)]_{M,n} + [ntf'(k) * q(k)]_{M,n}$ can be derived analytically by the Lagrange multiplier technique. Nevertheless, this method is based on the hypothesis that the quantization error can be considered as white noise and it is uncorrelated with signals, which is not true for an ADC with 1-bit quantizer (in our circuit). In addition, these kind of digital filters are not able to correct the conversion error including noise and distortion due to the circuit non-ideality.

An optimal filtering algorithm is proposed in [83] which allows reducing overall quantization error by taking into account the order of occurrence of 1 or -1 in modulator output bit sequence. A counter is usually used for reconstructing the modulator input, the estimation of the modulator input \hat{x} is calculated by:

$$\hat{x} = \frac{\sum_{i=1}^n b(i)}{n} \quad (2.12)$$

where n is the number of clock cycles and $b(i)$ is the modulator output at cycle i . This algorithm only takes into account the average of these n -bit modulator output sequence. However the modulator output sequence $b(1), b(2), \dots, b(i)$ can indicate the limit of x as following illustration. In

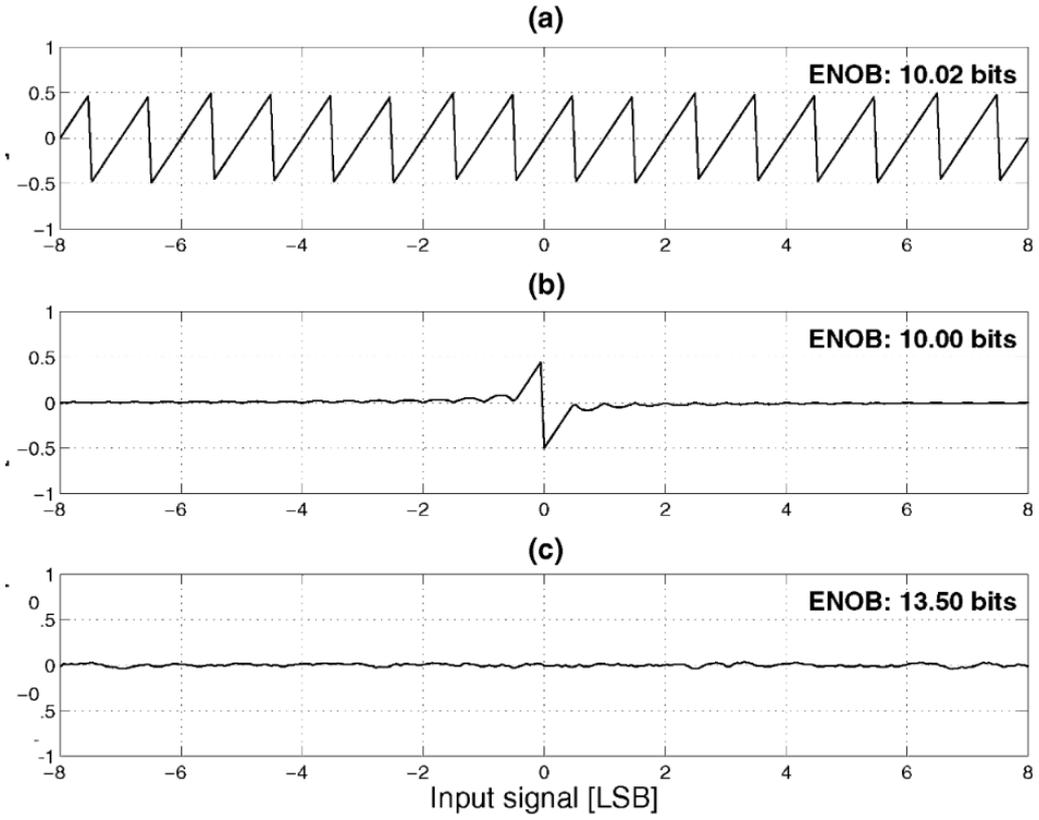


Figure 2.20: Quantization error of a 10-bit first-order converter around zero input. (a) Using first-order digital filter. (b) Using second-order digital filter (two integrators). (c) Using second-order filter with dither signal injected into the loop

cycle i , the integrator output $y(i)$ can be expressed as:

$$y(i) = ix - \sum_{j=1}^i b(j) \quad (2.13)$$

and the modulator output $b(i)$ is decided by:

$$b(i) = \begin{cases} 1 & y(i) > 0 \\ -1 & y(i) < 0 \end{cases} \quad (2.14)$$

So according to the value of $b(i)$, in each cycle, we can conjecture $\frac{\sum_{j=1}^i b(j)}{i}$ is an upper or a lower limit of x . We define x_{min} is the biggest value among the lower limit values while x_{max} is the least value among the upper limit values until cycle n . Finally, the modulator input can be estimated as:

$$\hat{x}' = \frac{x_{max} + x_{min}}{2} \quad (2.15)$$

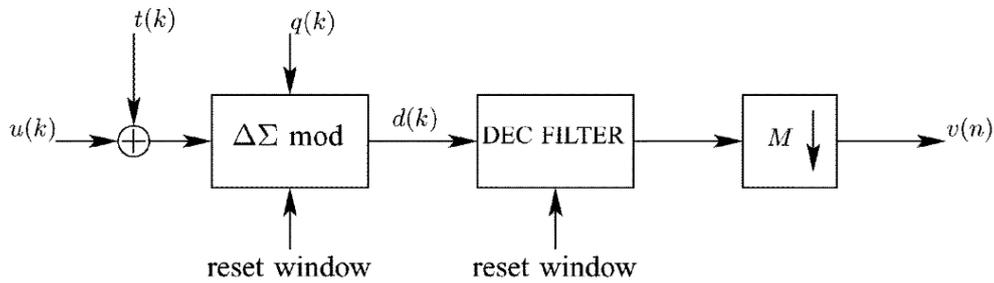


Figure 2.21: Block diagram of a typical $I\Sigma\Delta$ ADC with thermal noise at the input.

This algorithm can better bind the estimated modulator input compared to the counter. As a result, it reduces the quantization error. The optimal filtering algorithm can also be adapted for other cases: first-order $I\Sigma\Delta$ ADCs with sweep input[84] or second-order $I\Sigma\Delta$ ADCs[84] or even MASH $I\Sigma\Delta$ ADCs[84]. However the key of this optimal filtering is to know exactly the expression of the comparator input. In other words, the optimal filtering algorithm works well for an ideal $I\Sigma\Delta$ ADC, however it is not suitable for a non-ideal one unless we can identify precisely the defaults. For this reason, it has never been applied in a real circuit as it cannot resist noise.

2.4 Conclusion

After decades of research, the $\Sigma\Delta$ ADCs are well developed. Significant improvement in accuracy and conversion speed are achieved by applying new $\Sigma\Delta$ ADC structures. Meanwhile, in the company of diverse $\Sigma\Delta$ ADC structures, some calibrations techniques for $\Sigma\Delta$ ADC are also developed to compensate the degradation due to circuit imperfections. However, these calibration methods can be hardly applied in our circuit as we employ a one-bit quantizer. In addition, we can scarcely find calibrations techniques for $I\Sigma\Delta$ ADCs but some optimizations of digital reconstruction filters which only for ideal $I\Sigma\Delta$ ADCs. So it is a challenge to find a calibration or even a correction method for Pieretta.

Chapter 3

Two-Step $\Sigma\Delta$ ADC and Circuit

Pieretta

A chip of $\Sigma\Delta$ modulator has been designed by our team and it is called Pieretta. All my research revolves around Pieretta, including comprehension of its architecture, understanding design details and its calibration. Before the beginning of my PhD, the $\Sigma\Delta$ modulator architecture has been defined and the analog circuit design has been designed by a previous PhD student: Pierre Bisiaux. Then I designed the digital circuit. Since the objective of my research is to develop a calibration method, it is indispensable to figure out the characteristics of the $\Sigma\Delta$ modulator. For this consideration, the modulator has been designed as programmable through the digital circuit so that we could determine the modulator parameters maximally in the test.

This chapter will introduce Pieretta from the following aspects: ADC modeling, circuit design and simulation results. In the first section, the mathematical model of the $\Sigma\Delta$ modulator will be given. In the next section, the circuit design is presented, including the analog and the digital parts. Although the analog part is not my work, I will also introduce it so that we can better understand the details of the operation sequence of the $\Sigma\Delta$ modulator. This is important for the illustrations in following chapters. At the end, the results of post-layout simulations are presented.

3.1 $\Sigma\Delta$ ADC Modeling

At first approach, $\Sigma\Delta$ ADCs can be considered as $\Sigma\Delta$ ADCs with reset. It generally includes a $\Sigma\Delta$ modulator and a digital reconstruction filter. Fig. 3.1 shows the architecture of an $\Sigma\Delta$ ADC where X is the analog input to be converted while D is the converted digital output. In a

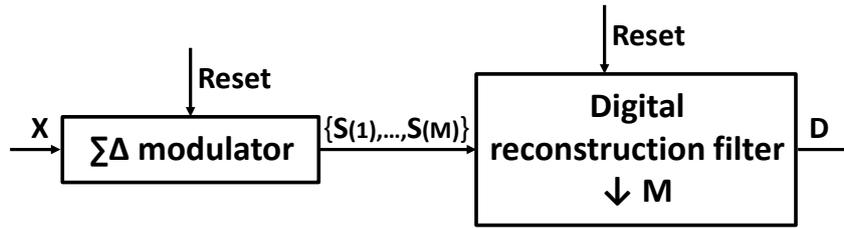


Figure 3.1: High level architecture of an IΣΔ ADC

conversion, the $\Sigma\Delta$ modulator converts an analog input to a set of digital outputs (either in the form of binary bits or multi-bit digital values) during a fixed number of clock cycles M . The digital reconstruction filter aims at interpreting the digital modulator outputs into a quantified value which is the estimation of the ADC input.

The principle of both $\Sigma\Delta$ ADCs and IΣΔ ADCs is $\Sigma\Delta$ modulation which is a method for encoding analog signals into digital signals. The difference between $\Sigma\Delta$ ADCs and IΣΔ ADCs is that there is no memory effect between conversions in an IΣΔ ADCs. Because of the absence of memory effect, IΣΔ ADCs can provide accurate estimation of ADC inputs in each individual conversion while $\Sigma\Delta$ ADCs focus more on the spectral characteristics of input signals. This character makes IΣΔ ADCs suitable for sensor applications such as biology, medicine and image.

No matter whether a sampling and hold block is employed or not in an IΣΔ ADC, the information of a DC input signal of the ADC is converted into two parts: the digital outputs of modulator (such as N_{out} in equation (2.1)) and the last value of the analog modulator intern signal known as the residue (such as $V_{out}(N+1)$ in equation (2.1)). In other words, an error-free estimation of the ADC input can be obtained if we know exactly this information. However, for classical IΣΔ ADCs, the residue is an analog value, so it is inaccessible. Generally, we ignore it and that results in the quantization error. It is easy to think that if the residue can be measured, we will be able to get a fine estimation of the ADC input. It is similar to the idea of pipeline ADCs. To do that, the second step conversion is required and the residue of the first step should be sampled and maintained during all along the second step. That is why a S/H is indispensable in our circuit.

There are diverse architectures of $\Sigma\Delta$ modulators such as high-order, MASH, multi-loop and multi-bit ones, all their objectives are high resolution. High-order single-loop modulators have better noise-shaping performance for quantization error but it may be unstable. The multi-bit quantizer is a solution for the instability, however it requires dynamic element matching techniques (DEM)[85] which requires additional current consumption for the additional blocks and increases the circuit complexity. Another approach for realizing higher-order $\Sigma\Delta$ modulators is

to use a cascade structure [86]. Although, it is possible to make the cascade of several first-order or second-order modulators or any other ADCs which is inherently stable for any order and over all regions of operation, the performance of cascaded modulators is more sensitive to the imperfections of the analog components than that of single-loop modulators. As the mismatch of noise transfer function (NTF) in digital reconstruction filter leads to noise leakage, it degrades the quantization error cancellation. In summary, the second-order modulator is a good compromise between resolution and circuit complexity. Among diverse architectures of second-order $\Sigma\Delta$ modulator, feed-forward topology is popular in recent papers [71][72][73], as it cancels the input signal at the input of loop filters so that they only have to process quantization noise. In addition to this, since there is no DC component in all integrator's output, the dynamic range requirements in all integrators are relaxed. Because of all the previous considerations, a second-order cascade integrator feed forward (CIFF) modulator architecture has been chosen for Pieretta.

3.1.1 Second-Order $\Sigma\Delta$ Modulator

As presented previously, a reset action occurs at the beginning of each conversion in an $\Sigma\Delta$ modulator. The architecture of our $\Sigma\Delta$ modulator is shown in Fig. 3.2. In order to adjust the range of the residue (whose definition will be introduced later) so that it can be as large as the range of the ADC input as possible, the values of coefficients are determined as $a_1 = a_2 = 0.5$, $a_3 = 2$.

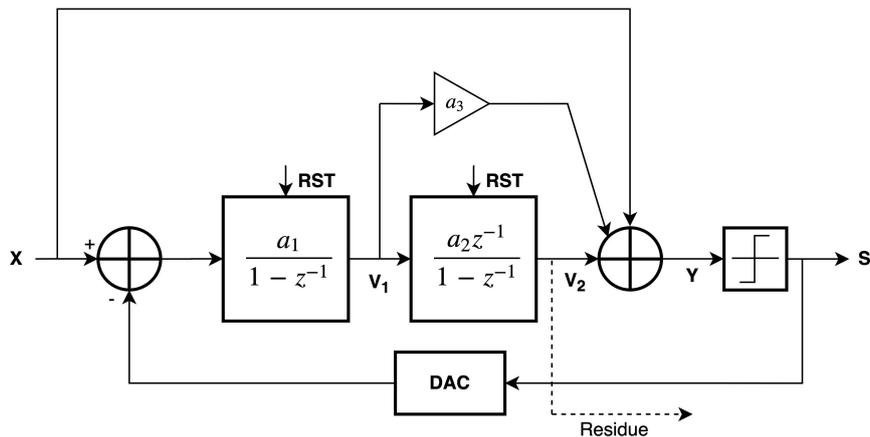


Figure 3.2: A second-order CIFF modulator

As the modulator is reset at the beginning of each conversion cycle and its input is sampled and maintained during the conversion, in time domain we have $V_1(0) = V_2(0) = 0$, and at the end

of cycle i , the two values of this two integrators can be expressed respectively as:

$$V_1(i) = V_1(i-1) + a_1(X - V_{ref}S(i)) \quad (3.1)$$

$$V_2(i) = V_2(i-1) + a_2V_1(i-1) \quad (3.2)$$

where X is the modulator input, and S is the binary output sequence of the comparator as well as the outputs of the modulator. Thus after M cycles, a modulator output bit sequence of M bits is obtained. By iterating equation (3.1) and equation (3.2), the value of the second integrator output is expressed as:

$$V_2(M) = a_1a_2 \sum_{j=1}^{M-1} \sum_{i=1}^j (X - V_{ref}S(i)) \quad (3.3)$$

where V_{ref} is the reference voltage in feedback. Arranging equation (3.3), the modulator input X can be expressed as:

$$X = \frac{2V_{ref}}{M(M-1)} \sum_{j=1}^{M-1} \sum_{i=1}^j S(i) + \frac{2}{a_1a_2M(M-1)} V_2(M) \quad (3.4)$$

The value of the second integrator at cycle M , $V_2(M)$, is called the residue. If M is great enough, the term involving $V_2(M)$ is negligible regarding X . Therefore, X can be estimated by:

$$\hat{X} = \frac{2V_{ref}}{M(M+1)} \sum_{j=1}^M \sum_{i=1}^j S(i) \quad (3.5)$$

And the quantization error

$$E_q = \frac{2}{a_1a_2M(M-1)} V_2(M) \quad (3.6)$$

is guaranteed below a very little value since the value of integrator is bound by $\pm V_{ref}$ as long as the input signal remains in a limited domain. In this case, the equivalent LSB quantization error E_{LSB} can be expressed as:

$$E_{LSB} = \frac{2V_{ref}}{a_1a_2M(M+1)} \quad (3.7)$$

and the effective number of bits (ENOB) can be estimated by:

$$\begin{aligned} ENOB &= \log_2\left(\frac{2V_{ref}}{E_{LSB}}\right) \\ &\approx \log_2\left(\frac{V_{in,max}}{V_{ref}} \cdot \frac{a_1a_2M^2}{2}\right) \end{aligned} \quad (3.8)$$

where $V_{in,max}$ is the input dynamic range. It should be noticed that the value of ENOB given by equation (3.8) is the value in worst case because E_{LSB} represents the worst quantization error. According to equation (3.8), to obtain a greater ENOB, M can be increased in order to reduce the contribution of the residue in the quantization error. However, with a linear increase of ENOB, M should increase in exponential way, as shown in Fig. 3.3. There is also another method: a second

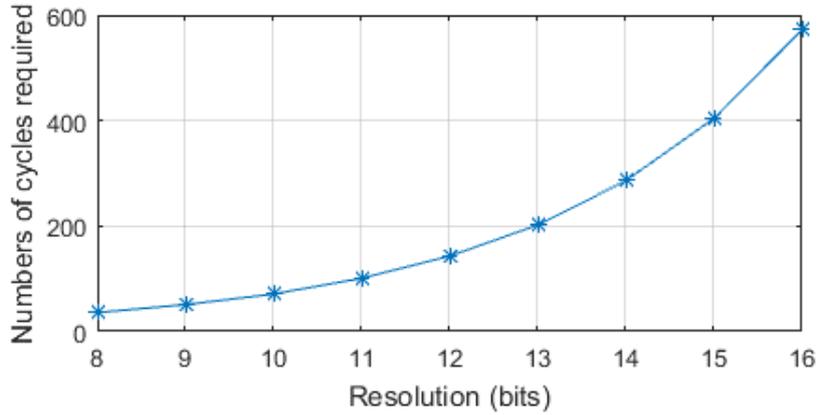


Figure 3.3: Numbers of clock cycles required in the function of ENOB

step can be added to measure this residue. That is actually the fundamental of the two-step ADCs[87].

3.1.2 Two-step conversion

The objective of carrying out a conversion over two steps is to get a higher resolution without a significant increasing of conversion time. For the first step, the modulator input is a sampled pixel output previously stored in a S/H circuit. Omitting the derivation details presented in 3.1.1, at the end of the conversion of the first step, we have:

$$X_1 = V_{pixel} = \frac{2V_{ref}}{M_1(M_1 + 1)} \sum_{j=1}^{M_1} \sum_{i=1}^j S_1(i) + \frac{2V_{21}(M_1)}{a_1 a_2 M_1 (M_1 + 1)} \quad (3.9)$$

where S_1 is the modulator output and M_1 is the number of clock cycles during the first step. $V_{21}(M_1)$ is the residue. The conversion in this step gives a coarse estimation of the pixel output. To get a fine estimation of the pixel output, the second step is taken to estimate the residue. After the M_2 -cycle conversion of the second step, we have :

$$X_2 = V_{21}(M_1) = \frac{2V_{ref}}{M_2(M_2 + 1)} \sum_{j=1}^{M_2} \sum_{i=1}^j S_2(i) + \frac{2V_{22}(M_2)}{a_1 a_2 M_2 (M_2 + 1)} \quad (3.10)$$

where S_2 is modulator output and $V_{22}(M_2)$ is the value of the second integrator after M_2 clock cycles in the second step. Replacing $V_{21}(M_1)$ in equation (3.9) by (3.10), we obtain:

$$V_{pixel} = \frac{2V_{ref}}{M_1(M_1 + 1)} \sum_{j=1}^{M_1} \sum_{i=1}^j S_1(i) + \frac{4V_{ref}}{a_1 a_2 M_1(M_1 + 1)M_2(M_2 + 1)} \sum_{j=1}^{M_2} \sum_{i=1}^j S_2(i) + \frac{4V_{22}(M_2)}{(a_1 a_2)^2 M_1(M_1 + 1)M_2(M_2 + 1)} \quad (3.11)$$

Based on equation (3.11), V_{pixel} can be considered as an estimation term plus a new term of quantization error E_{q2} :

$$V_{pixel} = \widehat{V}_{pixel} + E_{q2} \quad (3.12)$$

where the estimation term is composed of the functions of S_1, S_2

$$\widehat{V}_{pixel} = \frac{2V_{ref}}{M_1(M_1 + 1)} \sum_{j=1}^{M_1} \sum_{i=1}^j S_1(i) + \frac{4V_{ref}}{a_1 a_2 M_1(M_1 + 1)M_2(M_2 + 1)} \sum_{j=1}^{M_2} \sum_{i=1}^j S_2(i) \quad (3.13)$$

and

$$E_{q2} = \frac{4V_{22}(M_2)}{(a_1 a_2)^2 M_1(M_1 + 1)M_2(M_2 + 1)} \quad (3.14)$$

Thus the new equivalent LSB can be recalculated as:

$$E_{LSB2} = \frac{4V_{ref}}{a_1 a_2 M_1(M_1 + 1)M_2(M_2 + 1)} \approx \frac{4V_{ref}}{a_1 a_2 M_1^2 M_2^2} \quad (3.15)$$

And the effective number of bits for the two-step conversion is given by:

$$ENOB_2 = \log_2\left(\frac{2V_{ref}}{E_{LSB2}}\right) \approx \log_2\left(\frac{V_{in,max}}{V_{ref}} \cdot \frac{(a_1 a_2)^2 M_1^2 M_2^2}{4}\right) \quad (3.16)$$

For the same value of total number of clock cycle M ($M = M_1 + M_2$), when $M_1 = M_2$, $ENOB_2$ has its maximum value. Fig. 3.4 shows the comparison of the global numbers of clock cycles M required for the one-step conversion and the two-step conversion. According this Fig. 3.4, for the same resolution, the two-step conversion largely reduces conversion time compared to the one-step conversion one. Since equation (3.5) and equation (3.13) have well shown the expressions of estimation of the modulator input respectively in one-step mode and two-step mode, what we

need now is to implement these algorithms in digital filters which will be presented in the following.

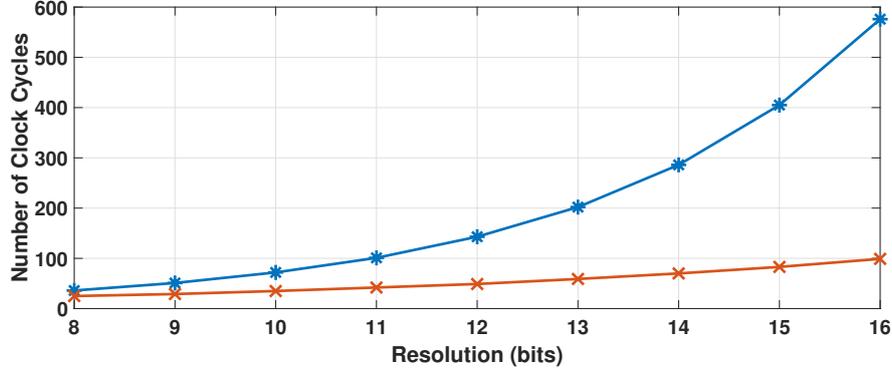


Figure 3.4: Comparison of numbers of clock cycles required in one-step and two-step conversions

3.1.3 Digital Reconstruction Filter

A digital filter is required in order to transfer the modulator output bit sequence to a quantified value. The output of the digital reconstruction filter is the final estimation of the ADC input. For one-step conversion, equation (3.5) can be used for estimating the ADC input:

$$\widehat{V}_{pixel} = \widehat{X} = V_{ref} \frac{2}{M(M+1)} \sum_{j=1}^M \sum_{i=1}^j S(i) \quad (3.17)$$

For a two-step conversion, equation (3.13) can be used for estimating the ADC input, and it can also be written as:

$$\begin{aligned} \widehat{V}_{pixel} &= \frac{2V_{ref}}{M_1(M_1+1)} \sum_{j=1}^{M_1} \sum_{i=1}^j S_1(i) \\ &+ \frac{4V_{ref}}{a_1 a_2 M_1(M_1+1) M_2(M_2+1)} \sum_{j=1}^{M_2} \sum_{i=1}^j S_2(i) \\ &= k_1 D_1 + k_2 D_2 \end{aligned} \quad (3.18)$$

where d_1 and d_2 share the same expression:

$$D_k = \frac{2}{M_k(M_k+1)} \sum_{j=1}^{M_k+1} \sum_{i=1}^j S_k(i), \quad k = 1, 2 \quad (3.19)$$

where $k_1 = V_{ref}$, $k_2 = \frac{2V_{ref}}{a_1 a_2 M_1(M_1+1)}$.

So algorithms to reconstruct the ADC input (in both one-step and two step conversions) can be realized by a FIR filter or a dual integrator. And each term of D_k can be seen as the output of the FIR, whose transfer function in z domain is:

$$F_{th}(z) = \frac{2}{M_k(M_k + 1)}(z^{-1} + 2z^{-2} + \dots + (M_k - 1)z^{-M_k+1}) \quad (3.20)$$

To distinguish it from other digital reconstruction filters introduced later, we call this filter "theoretical filter".

3.2 Design of I $\Sigma\Delta$ Modulator

As the modulator outputs are binary bits, post-processing can be performed off line on data processing software. Therefore, only the I $\Sigma\Delta$ modulators have been implemented in Pieretta while the algorithm of digital reconstruction filter will be tested on Matlab. In the circuit of I $\Sigma\Delta$ modulator, there are also the analog circuit and the digital circuit. The analog circuit at transistor-level has been designed by a previous PhD student. The transistors have been dimensioned with an approach of gm/Id[88]. Understanding the details of the analog circuit and completing the digital control circuit are steps before the research of calibration techniques.

3.2.1 Analog Circuit Part

Modulator Schematic and Operation Sequences

Generally, a low-pass single-bit $\Sigma\Delta$ modulator includes integrators and a comparator. In addition to this, an adder is required in the CIFF architecture. In particular, our I $\Sigma\Delta$ modulator performs a two-step conversion so a S/H block is employed to sample the residue at the end of the first step and maintain it during all along the second step. Of course the S/H is also used in the first step to sample and maintain the voltage of pixel. Fig. 3.5 shows the schematic of the analog circuit of the I $\Sigma\Delta$ modulator. The I $\Sigma\Delta$ modulator is controlled by two main signals: p1 and p2. And other signals are derived from p1 and p2 by adding different delays, such as p1. Fig. 3.6 shows the timing of the command signals p1 and p2 and operation sequences of the components of the I $\Sigma\Delta$ modulator. We can see that both integrators operate in a complementary way: when the first integrator works in the phase of sampling (phase a), the second one is in the phase of integration (phase b); when the first integrator executes integration (in phase b), the second one is in the phase a where it samples the value of the first integrator. Each cycle, a comparison occurs at the

end of phase b of the second integrator and just slightly before phase b of the first one. So for a conversion of M clock cycles, $S(M)$ is decided at the end of the M^{th} cycle. However the second integrator integrates at the beginning of $(M + 1)^{th}$ cycle.

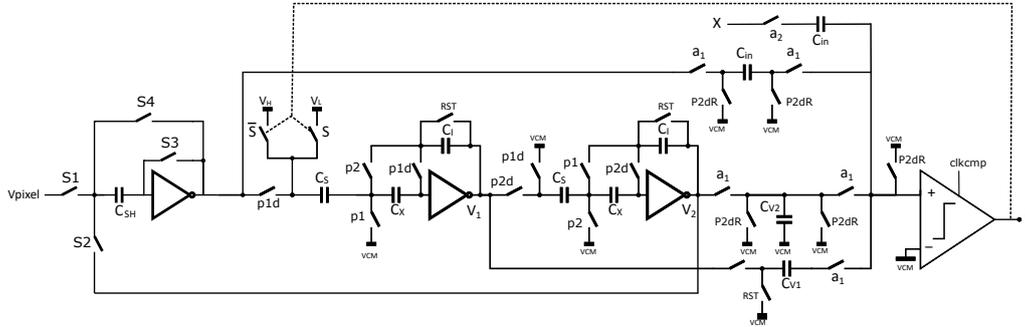


Figure 3.5: Schematic of the $\Sigma\Delta$ modulator

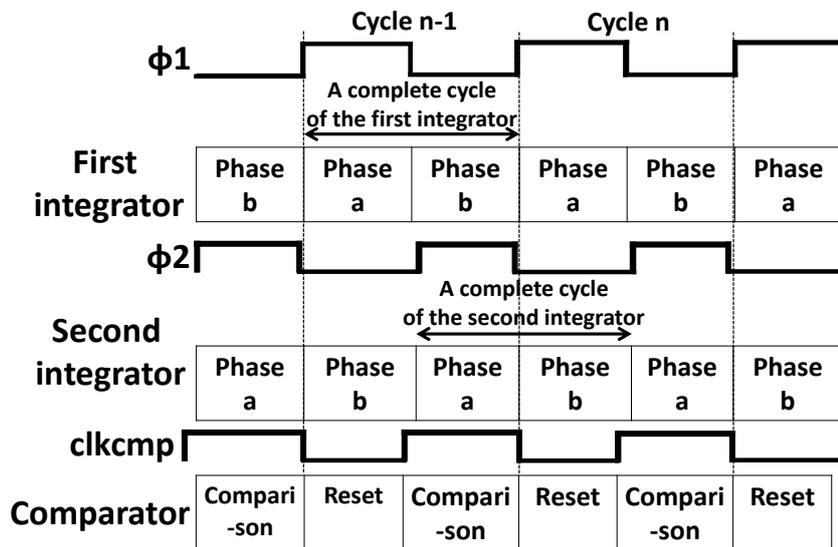


Figure 3.6: Timing diagram of the command signals and corresponding states of components

Inverter-Based Integrator with Auto-Zeroing

Fig. 3.7 shows the schematic of an ideal inverter-based SC integrator with auto-zeroing (black part). In order to cancel the offset (red part), the capacitor C_x ($C_x = C_i$) is added. The operation of an inverter-based SC integrators and its offset cancellation are explained as following. The integrator performs in two steps: the phase of sampling (phase a) and the phase of integration (phase b). During phase a, the integrator input V_i is sampled in C_s and on the other hand, one plate of C_i is switched to the amplifier input. This feedback connection forms a negative unity gain of V_{of} which is the offset voltage at the amplifier input. Therefore, its contribution to the amplifier

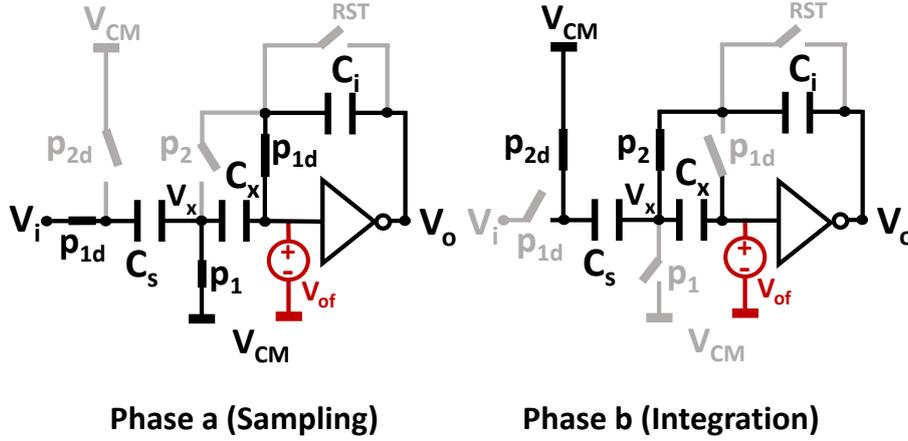


Figure 3.7: Schematic of inverter-based SC integrator

output is $-V_{of}$. So in phase a, integrator input and output at cycle n is given by:

$$V_{o,a}(n) = V_{o,b}(n-1) - V_{of} \quad (3.21)$$

At the same time, V_{of} is sampled in C_x and we have $V_{x,a} = V_{cm}$, where V_{cm} is the common mode voltage of the circuit, for the potential at node X in phase a. During phase b, node X is disconnected from ground and one plate of C_i is switched to node X . Since no charge passes through the amplifier input, the charge in C_x remains constant. Thus we still have $V_{x,b} = V_{cm}$, for the potential at node X in phase b. As a consequence, all charge in C_s is transferred to C_i . Besides, as the potential of one side of C_i shifts from V_{of} to V_{cm} , the contribution of V_{of} to amplifier output is canceled. In phase b, the relationship between the integrator input and output at cycle n is given by:

$$V_{o,b}(n) = V_{o,a}(n) + \frac{C_s}{C_i} V_i(n) + V_{of} \quad (3.22)$$

The final integrator output (V_o) is the value of integrator in phase b ($V_{o,b}$) and its expression is:

$$V_o(n) = V_o(n-1) + \frac{C_s}{C_i} V_i(n) \quad (3.23)$$

Gain-Boosting Inverter as an Amplifier

Amplifier is important component of both S/H block and integrators. In the context of image sensors where the width of the ADC is an important criterion, a saving in size on the amplifiers is to be achieved. In order to reduce the size of the amplifiers, architectures of inverter-based amplifiers were introduced by Chae [89]. In order to enhance the amplifier gain, architectures of

cascade inverters[90] or even gain-boosting have been created[91]. According to the analysis of Pierre Bisiaux in [92], a gain of 80 dB is sufficient to achieve 14 bits. For this architecture, the

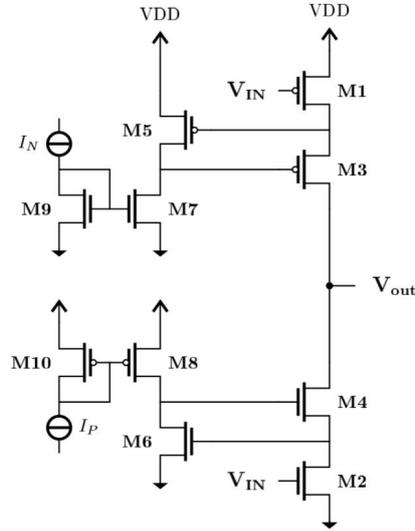


Figure 3.8: Schematic of gain-boosting inverter-based amplifier

inverter-based amplifier can operate in class-AB or class-C depending on the supply voltage. For class-AB, supply voltage needs to meet the conditions: $|V_{Tp}| + |V_{Tn}| < V_{dd}$. In this situation, transistors are in medium or strong inversion that gives a non-zero static current which can be significant, increasing the speed of the inverter. However, in strong inversion, the amplifier gain is relatively low. In our application, as the system clock is 20 MHz, a strong current is required in order to ensure a sufficient GBW and make the slew rate at least ten times greater than the system clock frequency. In order to use a minimum area to deliver this high current, an amplifier in class-AB is therefore used. For this type of inverter, it is easy to obtain a GBW of 100 MHz, a sufficient value to satisfy the time constraints of the system[92].

Combining the above requirements of speed and gain in terms of amplifier performance, an architecture operating in class-AB and using a boosting gain is used. This architecture is shown in Fig. 3.8, and in the condition $I_N = I_P = 1\mu A$, the dimensions of transistors are shown in Table 3.1. M1 to M4 are polarized as medium inversion to strong inversion in order to have a stable and relatively little gm as it is at the first stage of the cascaded amplifier, is gm is too large, the second stage risk to be saturated. While the "low-Vth" transistors M5 and M6 are in weak inversion in order to enhance the amplifier gain. M7 to M10 play a role to provide the current $I_N = I_P = 1\mu A$. The use of inverter-based amplifiers is one of the major distinguishing feature in this ADC design.

Transistors	W/L,	Gm/Id
M1	20/0.36	7
M2	5/0.36	7
M3	30/0.36	7
M4	7.5/0.36	7
M5	8/0.36	20
M6	2/0.36	20
M7,M9	1/0.36	15
M8,M10	3/0.36	15

Table 3.1: Dimensions of transistors in the gain-boosting inverter

Sampling-and-Hold Block

It is necessary to carry out a S/H block to maintain the modulator input constant during the conversion in each step. The inverter-based amplifier designed for integrators has sufficient characteristics in terms of gain or GBW because the input of S/H is DC voltage. So the inverter-based amplifier can also be used in the S/H circuit. The schematic in different states of the S/H block with auto-zeroing technique is shown in Fig. 3.9, where the command signals' chronograph is shown in Fig. 3.10. For the first step, when $S1$ and $S3$ are in state ON, the value of pixel is sampled. Then $S1$ and $S3$ become OFF and $S4$ is ON, during this time, the output of the S/H is maintained constant and it is equal to the value of pixel. While in the second step, $S2$ and $S3$ are first ON to sample the residue then they are switched off and $S4$ is switched on so that the S/H output remains constant during all along the second step conversion.

Adder and Comparator

In any second-order modulator with feed-forward, an adder is required to sum the X , V_1 and V_2 in front of the comparator. It is a passive adder composed of capacitors and switches. The architecture of the adder-comparator is shown in Fig. 3.11 and the chronograph of the command signals is shown in Fig. 3.12. a_1 and a_2 are derivative of p_1 while a_2 is only enabled at the first clock cycle of each step to compare the modulator input (X) to the common voltage (V_{cm}) directly. On the opposite, $clkcmp$ and $p2dR$ are derivative of p_2 .

We chose a double-tail comparator. This architecture is widely used in imagery systems for its low consumption and its low kickback. The schematic of the comparator is shown in Fig. 3.13 with the dimensions of transistors shown in Table 3.2. When $clkcmp = 0$, $In+$ and $In-$ are both connected to V_{cm} and values of $OUT+$ and $OUT-$ are set as V_{cm} . While a decision is made when $clkcmp$ changes to 1. At this moment, the bottom $M0$ switch is closed to allow current to

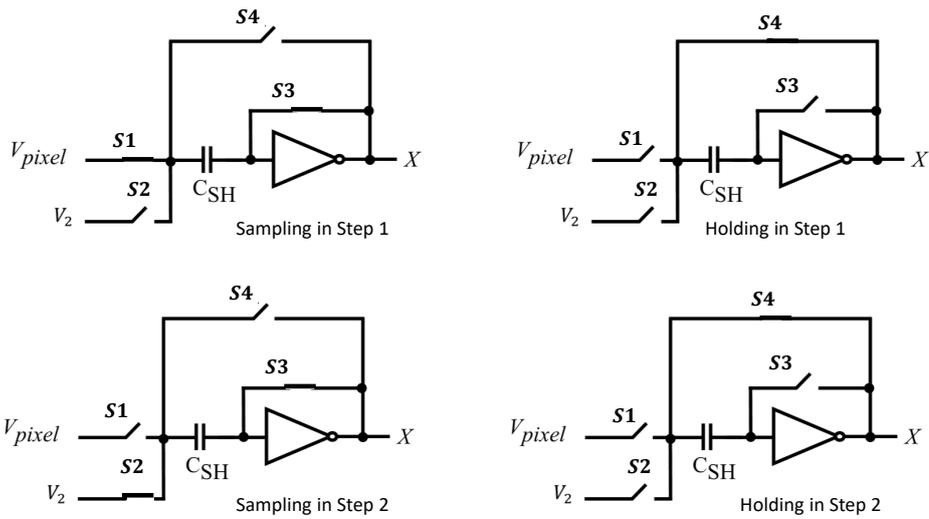


Figure 3.9: Schematic of Sampling-and-hold block

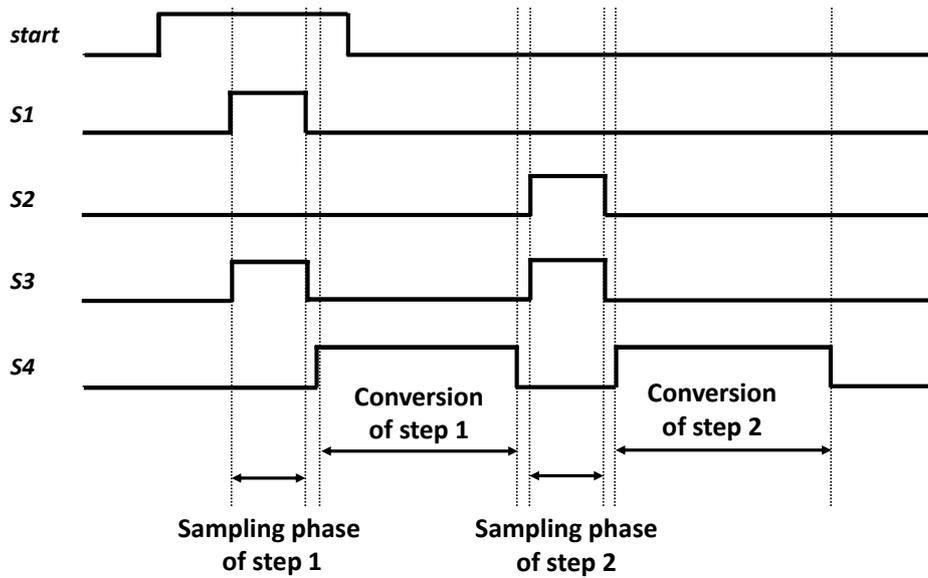


Figure 3.10: Timing diagram of the signals commanding S/H block

flow through the input pair. if a difference of voltage appears between I_{n+} and I_{n-} , a difference of current is created in branches, making a double reversed decision of the comparison.

Switches

In analog circuit, switches should be designed carefully to avoid the distortion of the analog signal. An NMOS or a PMOS is a simple switch. However, when this switch is opened, the charge present in the channel of drain-source will be distributed over the elements on both sides of the

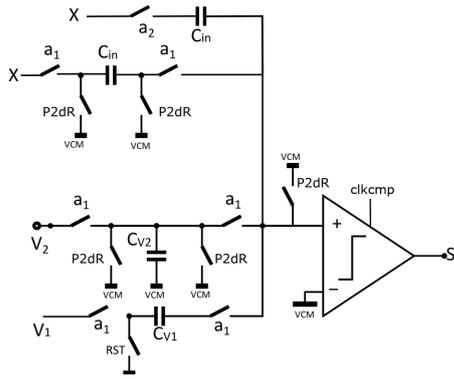


Figure 3.11: Schematic of adder-and-comparator

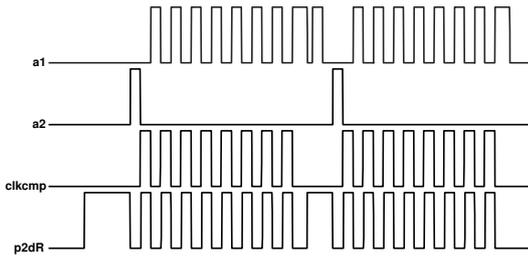


Figure 3.12: Timing diagram of command signals in adder and comparator

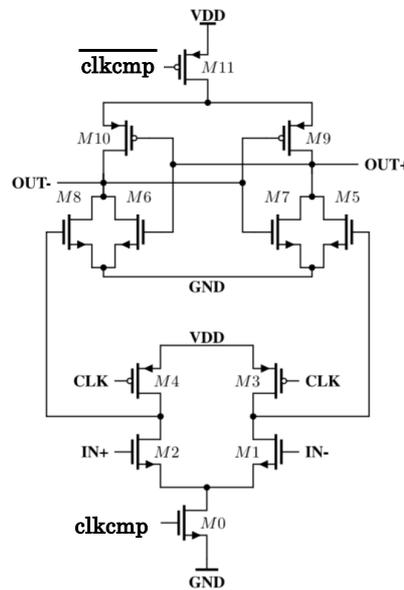


Figure 3.13: Schematic of double-tail comparator

Transistors	W/L
M1	8/0.18
M2	8/0.18
M3	1/0.18
M4	1/0.18
M5,M7,M9	0.56/0.18
M6,M8,M10	0.56/0.18

Table 3.2: Dimensions of transistors in the comparator

switch. This phenomenon is called charge injection. It brings an error to the analog signals in the modulators. In order to minimize this charge injection, dummy transistors are then added on the source and drain side, in order to absorb the charge. In order to reduce the channel resistance, "low-Vt" CMOS switches are used. For the same channel resistances, these transistors

Transistors	W/L
NMOS	0.44/0.18
PMOS	1/0.18

Table 3.3: Dimensions of transistors in switch

are smaller, allowing for easier integration. In order to obtain a linear resistance above all the transmission range, the dimension of PMOS transistors is larger than that of CMOS transistors. The dimension of transistors are summarized in Table 3.3. To be noticed that all the schematics of switches and dimensions of the transistors in the circuit are the same.

Non-overlap block

As two integrators in the modulator are forced to operate in a complementary by commanding the switches in the modulator with p1 and p2 and their derivatives with different delay, p1 and p2 must not be active at the same time so as not to make certain switches closed at the same time. If so, there will be short circuit. A commonly adopted solution is to use non-overlapping block to add a slight delay to p2, so that p2 start rising after p1 drops to level-low completely. The schematic diagram of non-overlapping block is shown in Fig. 3.14.

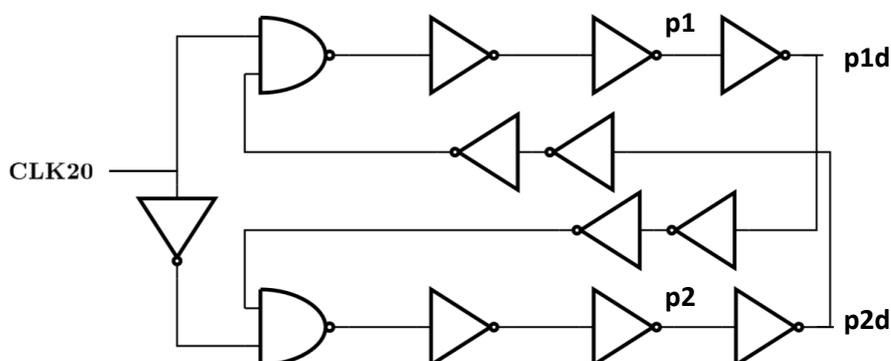


Figure 3.14: Schematic of non-overlap block

3.2.2 Digital Circuit

The digital circuit in the $\Sigma\Delta$ modulator serves to generate signals to control the switches in the analog circuit so that the $\Sigma\Delta$ modulator can work correctly. It includes two components: a state machine and a non-overlap block.

Considering the circuit characterization in the testing session after tape-out, we designed the circuit to be programmable. With different parameter settings, we can make the $\Sigma\Delta$ modulator work in diverse conversion modes with different configuration. This idea is realized by a state machine coded in Verilog. It generates enable signals according to the parameter settings.

The non-overlap block receives the enable signals from the state machine then it generates the command signals with proper delay to the analog circuit.

Conversion Modes and Configurations

There are two input control signals and a external clock : rst_sys , $start$ and $clk40$. The modulator is reset immediately when $rst_sys = 1$ while a conversion is triggered by $start$. The external clock, $clk40$, is divided by 2 in digital circuit.

The $\Sigma\Delta$ modulator is able to perform conversions in two modes: the default mode and the programmable mode. In the default mode, the $\Sigma\Delta$ modulator is supposed to convert over two steps with 36 clock cycles in each step. While in the programmable mode, the parameters of $\Sigma\Delta$ modulator, namely the number of steps, the length of sampling phase, the number of clock cycles in each step, can be configured.

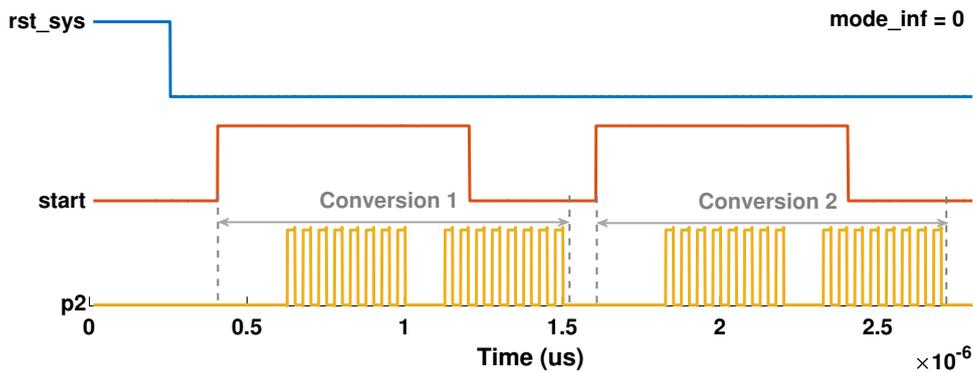
To distinguish the default mode and the programmable mode, we define a one-bit variable $Mode_prog$. By setting the 1-bit parameter $Mode_prog$ as 0 through interfaces, the $\Sigma\Delta$ modulator functions in the default mode. While when $Mode_prog$ is set as 1, the ADC is in programmable mode.

There are another three variables to configure the parameters of conversion. They are: NB_step , NB_samp and OSR . Although the ADC is supposed to perform conversions in two-step mode, we can also make it work in one-step, three-step even in four-step conversion modes by setting the value of the 2-bit parameter NB_step from 0 to 3. It is possible to choose the number of clock cycles for all the steps through the parameter of 6 bits, OSR . The number of clock cycles is decided by $4 \times (OSR + 1)$. There are also possibility to change the length of sampling phase (the number of clock cycles taken in sampling phases) by setting the value of the 2-bit parameter NB_samp and the length of sampling phase varies from 1 clock cycles to 4 clock cycles.

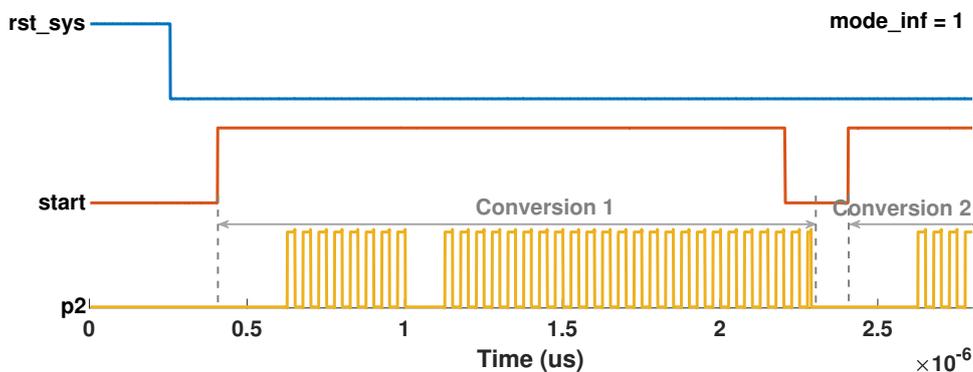
To be mentioned that in the programmable mode, there are two sub modes: the finite mode and the infinite mode. The differences between them are the way to trigger the $\Sigma\Delta$ modulator and the way to stop a conversion. In the finite mode, during the standby mode, the $\Sigma\Delta$ modulator reads the input $start$. If $start = 1$, it reads and registers the values of parameters and launches a new conversion as configured. After finishing a conversion, the ADC returns to the standby mode and keeps testing $start$ until it becomes 1. In this mode, if $start$ is always kept in high voltage

level, conversions are triggered automatically one by one. The numbers of clock cycles in each step are equal and the maximum value is limited to 256. While in the infinite mode, if $start = 1$, the modulator will continue converting permanently. It means if the period of $start = 1$ is long enough, the $\Sigma\Delta$ modulator performs its preceding steps as configuration and loops in the last step without stop until $start$ change from 1 to 0. Then the $\Sigma\Delta$ modulator returns to the standby mode and waits for the next $start$ in high voltage level. In this case, the number of clock cycles of the last step in a conversion can be set as we want by applying the $start$ with a proper length. We use two parameters to describe the configuration in this mode: M for the global number of clock cycles and M_1 for that of every step except the last one.

The variable $Mode_inf$ is reserved to configure the sub-modes. When $Mode_inf = 0$, the $\Sigma\Delta$ modulator perform in the finite mode while if $Mode_inf = 1$, it performs in the infinite mode. Fig. 3.15a and Fig. 3.15b show examples in programmable mode respectively when $Mode_inf = 0$ and $Mode_inf = 1$, in both cases, we configure $OSR = 1$ ($M_1 = 8$), $NB_step = 1$ (two steps) and $NB_samp = 0$ (one clock cycle). Table 3.4 shows the summary of all conversion modes.



(a) Diagram of timing in finite mode



(b) Diagram of timing in infinite mode

Figure 3.15: Examples of timing in finite mode and infinite mode

Default mode: $Mode_prog = 0$	
ADC operates in two steps with $M_1 = M_2 = 36$ taking 1 clock cycle to sample in each step	
Programmable mode: $Mode_prog = 1$,	
Finite mode: $Mode_inf = 0$	Infinite mode: $Mode_inf = 1$
Number of steps, number of clock cycles and length of sampling in each step of the ADC only depend on NB_step , OSR , NB_samp . The ADC will finish a complete conversion once it is triggered.	The number of steps, the number of clock cycles and the length of sampling in each step of the ADC depend on NB_step , OSR , NB_samp and $start$. The ADC stop converting once $start$ changes to 0.

Table 3.4: Descriptions of conversion modes with different configurations

State Machine

The state machine serves to read input parameters and generate enable signals to non-overlap block according parameter configuration as well as $clk20$ which is the external clock $clk40$ divided by 2. The diagram of its states is shown in Fig. 3.16. Descriptions of each states and transfer conditions are shown in Table 3.5 and Table 3.6.

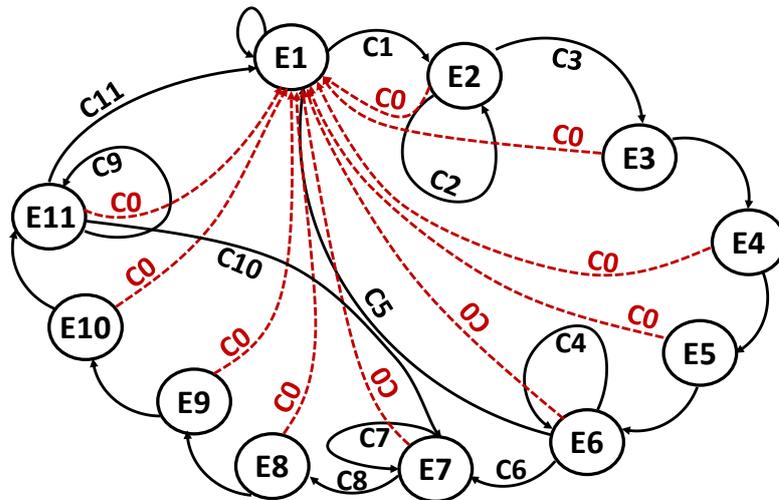


Figure 3.16: Diagram of states

In brief, state machine controls the modulator to complete a series of actions as: sampling, converting, finishing conversion or going into next step.

3.3 Post-layout simulation results

With a complete design of the $\Sigma\Delta$, simulations in post-layout have been perform. It allowed estimating the original performances of the modulator and characterizing the circuit to the greatest

States	Descriptions
E1	Standby mode
E2	Sampling phase of the first step
E3	Gap clock cycle between sampling phase and conversion phase
E4	The first $p1$ of the first step.
E5	The first $p2$ of the first step. During this period, only the modulator input is compared to V_{cm}
E6	The rest of clock cycles of the first step.
E7	Sampling phase of the second step
E8	Gap clock cycle between sampling phase and conversion phase
E9	The first $p1$ of the second step.
E10	The first $p2$ of the second step. During this period, only the modulator input is compared to V_{cm}
E11	The rest of clock cycles of the second step.

Table 3.5: Descriptions of states

Labels	Descriptions	Conditions
C0	Reset or conversion stops in infinite mode.	$rst_sys = 1$ or ($mode_inf = 1$ and $start = 0$)
C1	Conversion begins.	$rst_sys = 0$ and $start = 1$
C2	Sampling1 doesn't finish 1.	$rst_sys = 0$ and $cpt_samp1 < NB_samp$
C3	Sampling1 finishes.	$cpt_samp1 = NB_samp$
C4	The first step conversion doesn't finish .	$rst_sys = 0$ and $cpt_osr1 < OSR$
C5	One-step Conversion finishes.	$cpt_osr1 = OSR$ and $NB_step = 0$
C6	Multi-step conversion.	$NB_samp > 0$
C7	Sampling2 doesn't finish 1.	$rst_sys = 0$ and $cpt_samp2 < NB_samp$
C8	Sampling2 finishes.	$cpt_samp2 = NB_samp$
C9	The current step in multi step mode doesn't finish .	$rst_sys = 0$ and $cpt_osr2 < OSR$
C10	The current step in multi step mode finishes, cpt_osr2 will be reset.	$cpt_osr2 = OSR$ and $cpt_step < NB_step$
C11	The multi-step conversion finishes.	$cpt_osr2 = OSR$ and $cpt_step = NB_step$

Table 3.6: Descriptions of transfer conditions

extent.

3.3.1 Equivalent resolution

We use Equivalent Resolution (ER) to represent the performance of the ADC in simulations in bits. For a set of ramp signal distributing uniformly in $[0, V_{in,max}]$, ER can be calculated using the following formula:

$$ER = \frac{10 \log_{10} \left(\frac{V_{in,max}^2}{3P_{error}} \right)}{6.02} \quad (3.24)$$

where P_{error} is the average of conversion error powers. All the ER in this paper are calculated using equation (3.24).

3.3.2 Parallel Simulations and Choice of the Modulator Input Values

We planned to test a set of inputs whose values varying between $\pm V_{in,max}$, where $V_{in,max}$ is the amplitude of the modulator input, in post-layout simulation. The number of samples should be sufficient large to ensure a stable estimation of ER , however, the post-layout simulation will take extremely long time for thousands of samples. We took two weeks for about 1500 samples in the post-layout simulation for one-step modes. In order to reduce the simulation time, we divided the whole simulation into four short simulations executed by different cores in parallel. And the input amplitude of each short simulations became $0.25V_{in,max}$. To avoid overload of integrator, $V_{in,max}$ was chosen as $0.8V_{ref}$.

It is simple to generate a set of ramp signals in the range of $[V_{min}V_{mas}]$ with constant step. However, there is a great inconvenience performing simulations with this kind of signals: we cannot stop simulations at whenever we want. To solve this problem, we generated the input signals in each short simulation in a new way. Furthermore, the inputs are correlated and do not change very much between each value. This may hide some non-idealities issues that appear with large input steps.

N loops should be taken to generate all the inputs. During each loops, the input values are coded by a definitive number of binary bits. One bit is used for the inputs generated in the first loop so there are two different values in the input set. Then two bits are used for those generated in the second loop and after the second loop, two more inputs of different values are added to the input set. In each loop, we add 1-bit precision to code the input values and the length of the input set doubles. So in the last loop, the precision of the input is N bits. The diagram of inputs generation is shown in Fig. 3.17. Fig. 3.18 shows an example of the inputs generated with this method.

3.3.3 Post-layout Simulation of One-Step Conversion

In order to understand the circuit characteristics, we first make the ADC execute the one-step conversion. From equation (3.8), a 14-bit resolution is guaranteed by applying 286 clock cycles in one conversion. In practice, we chose 400 clock cycles with some margin. And finally there were 5600 input samples. According the simulation of an ideal $\Sigma\Delta$ modulator model on Matlab, the ER of the one-step conversion with 400 clock cycles is 15.4 bits, as shown in Fig. 3.19 (a).

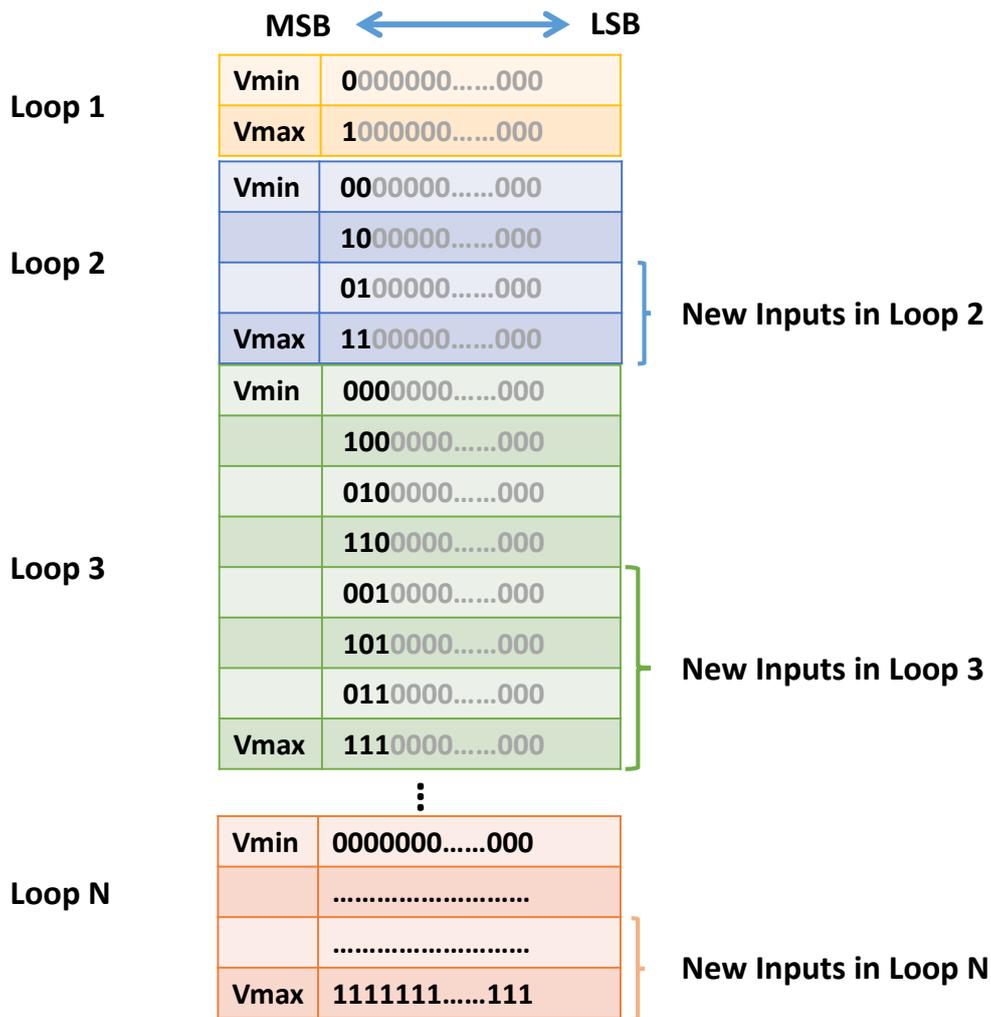


Figure 3.17: Diagram of the inputs generation.

In the post-layout simulation, the circuit is supplied by an ideal 1.8V DC voltage, driven by a 20 MHz clock. The ADC inputs and corresponding output bit sequences are recorded. For 5600 samples, the post-layout simulations took two weeks. The input reconstruction is done based on these post-layout simulation results in Matlab.

We use the theoretical filter to reconstruct the modulator inputs and obtain only $ER = 4.6$ bits. According to Fig. 3.19 (b), there is a significant gain and offset in conversion errors. This resolution is far away from the target. Considering that conventionally, the ADC resolution we talk about is the one after the correction of gain and offset. Therefore, we do this simple correction and obtain $ER = 12.3$ bits, as shown in Fig. 3.19 (c). To be noticed that the blue points are the samples for parameter estimation while the red points are samples for testing. The details of the correction of gain and offset will be presented in Chapter 4.

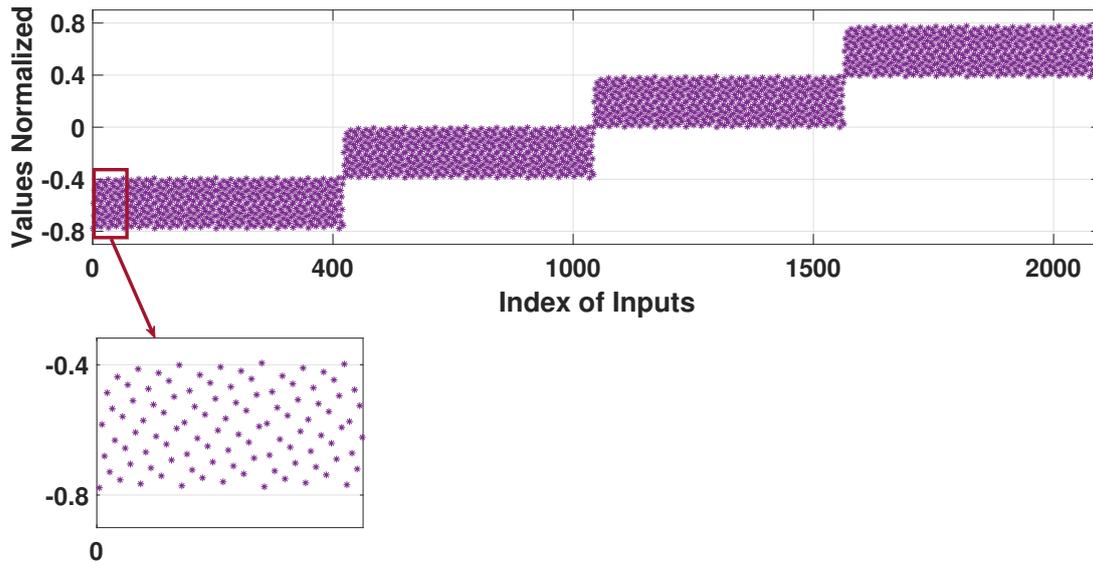


Figure 3.18: Inputs generated with new method.

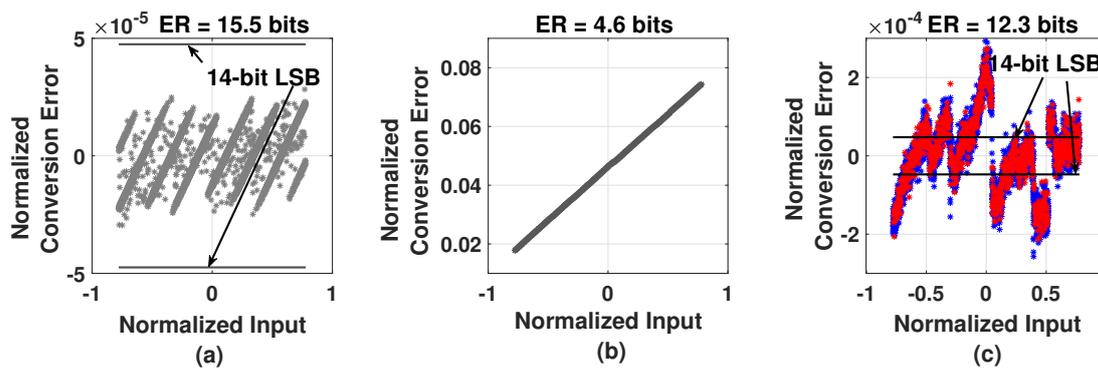


Figure 3.19: Equivalent resolution in one-step mode using the theoretical filter: (a) for the model of ideal ADC; (b) in post-layout simulation; (c) in post-layout simulation after the correction of gain and offset.

3.3.4 Post-layout Simulation of Two-Step Conversion

From equation (3.16), to guarantee a 14-bit resolution, 36 clock cycles are required for both steps. Simulations have been run for 2100 input samples. According the simulation of an ideal $\Sigma\Delta$ modulator model on Matlab, the ER of the two-step conversion with 36 clock cycles in both steps is 15.4 bits. Fig. 3.20 (a) shows the form of its quantization error. A post-layout simulation of two-step conversion has also been done in the same conditions.

Still using the theoretical filter, we obtain $ER = 4.8$ bits this time. Similar to the result in one step, ER is largely degraded by the gain and offset, as shown in Fig. 3.20 (b). After the correction of gain and offset, we finally obtain $ER = 12.0$ bits, as shown in Fig. 3.20 (c). These results are similar to those of the one-step conversion. Obviously, for both modes, resolutions are far away

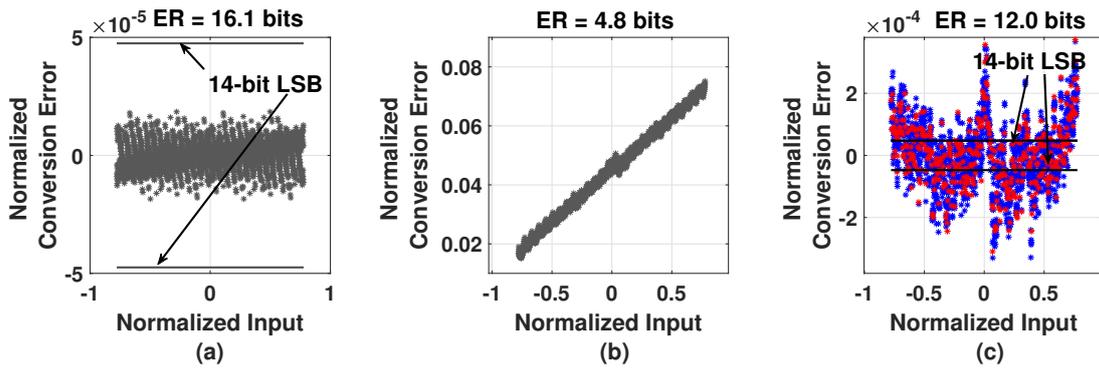


Figure 3.20: Equivalent resolution in two-step mode using the theoretical filter: (a) for the ideal ADC model; (b) in post-layout simulation; (c) in post-layout simulation after the correction of gain and offset.

from our target. Just completing the design of the $\Sigma\Delta$ modulator is not the destination, we are supposed to do more researches about calibration methods to make the $\Sigma\Delta$ modulator has the desired resolution.

3.4 Conclusion

In this chapter, we introduced mathematical model of $\Sigma\Delta$ modulator and illustrated its limit in the high-resolution applications. In order to achieve higher resolution without a large sacrifice of conversion speed, we proposed a two-step $\Sigma\Delta$ modulator. The second step is applied to estimate the residue so that we have a fine estimation of the ADC input. The proposed two-step $\Sigma\Delta$ modulator is able to provide a good trade-off between conversion speed and resolution.

The design of the analog circuit has been introduced briefly. In order to meet the size constraint of column-parallel ADCs, we employ inverter-based amplifiers with gain boosting in stead of conventional operational amplifiers. That is the main feature of the analog circuit.

We planned to make the parameters of the $\Sigma\Delta$ modulator configurable to the maximum extent so that we can study its characters after tape-out. This tentative idea was realized by the digital circuit design. In the control of the digital circuit, $\Sigma\Delta$ modulator can perform in several modes with different configuration.

Post-layouts simulations have highlighted grave degradation in resolution for both one-step and two-step conversions. Even after simply correcting gain and offset, ER still cannot achieve 14 bits. For this reason, calibration is required. The most important objective of my following research is to improve ER based on the values after the correction of gain and offset.

Chapter 4

$\Sigma\Delta$ ADC correction technique

From last chapter, we already know that in the post-layout simulations, the resolution of ADC is largely degraded. Therefore, a calibration is required.

Calibration techniques can be divided into two categories: foreground calibrations and background calibrations. Both consist on two tasks: a parameter identification phase and a correction phase based on the identified parameters. In a background calibration, the parameters are identified while the main function of the circuit is running. In a foreground calibration, during a learning phase, a learning signal is given at the input of the circuit and the corresponding outputs are recorded. The recorded outputs are then processed and a correction algorithm is built by minimizing the difference between the corrected signal and the expected one. In our case, a foreground calibration strategy is chosen because it is the most appropriate strategy for ADCs dedicated to imaging sensors.

Fig. 4.1 shows the block diagram of the calibration process. In order to validate the correction we proposed, we chose the following methodology: During a first phase, we present a learning

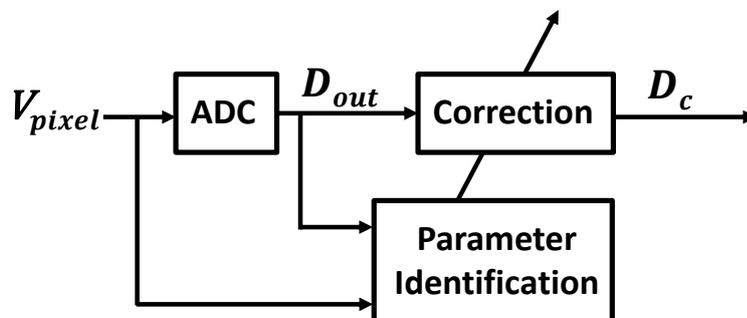


Figure 4.1: Diagram of calibration

signal at the input of the ADC and we record the output bits. The parameters are then determined by minimizing a reconstruction square error. Then in the second step we put a test signal at the input of the ADC and we record the output of the correction algorithm. The correction block outputs are compared to the ADC input signal in order to evaluate the reconstruction error. In order to guarantee the robustness of our techniques, we have ensured that the learning signal and the test signal are not the same. The learning signal is a regular sampling of a ramp voltage signal. The test signal is a sequence of random voltages, chosen with a uniform distribution in the input range. Also because of the large difference between the post-layout results and the transistor-level results, we guess that a correction based on the ideal model of $\Sigma\Delta$ ADCs is not sufficient and we chose to find out correction algorithms by considering the ADC as a black box.

In this chapter, we will first analyse, in a high level model, how the elementary conversion errors in each step degrade the global resolution. Then some state-of-art correction methods will be applied to post-layout simulation. Since the existing correction methods are not sufficient for our application, we propose some new correction methods. The new proposed correction methods will be presented and they will also be verified based on post-layout simulation. In order to make the proposed correction methods implementable, we go further to simplify them. The simplified correction methods and the associated results are shown. A correction method with the best trade-off between resolution and complexity will be selected for our two-step $\Sigma\Delta$ ADC. At the end, we will explore if the distribution of clock cycles between both steps has an impact on resolution.

4.1 Global error model

In each clock cycle, the modulator input X and its output S participate into the system loop. And after M clock cycles, we obtain a bit sequence S and a residue $V_2(M)$. Due to the circuit defects in the modulator (including S/H), we can no longer use equation (3.3) to express $V_2(M)$, but the following express:

$$V_2(M) = g(X, S) \quad (4.1)$$

where $g(X, S)$ is a non-linear function of X and S . If we isolate the linear term in X in $g(X, S)$ function, after arranging, we have:

$$X = f(X, S) + k(X, S)V_2(M) \quad (4.2)$$

where $f(X, S), k(X, S)$ are non-linear functions of X and S (except the linear term of X). According to the theory of I $\Sigma\Delta$ ADCs, $k(X, S)$ is negligible regarding 1 so that the impact of the residue is minimized in (3.4). To simplify in expression, we use k to replace $k(X, S)$ in the following illustration. Let us assume that this is still true in spite of the defaults. Then, if we use the theoretical filter to reconstruct X (equation (3.17)), equation (4.2) can be written in an equivalent way:

$$X = \widehat{X} + kV_2(M) + \epsilon \quad (4.3)$$

where ϵ contains non-linear terms of X and S . It indicates that an equivalent error term (ϵ) is added if we use the theoretical filter to reconstruct the input of a non-ideal ADC. In this situation, the total conversion error is the sum of $kV_2(M)$ and ϵ . ϵ is the accumulation of the errors of all cycles. So, we can guess that it much larger that the error term $kV_2(M)$. Therefore, the conversion error may exceed the LSB limit of the target resolution. This justifies why the theoretical algorithm is no longer sufficient for the input reconstruction for a one-step conversion mode.

For a two-step mode, after the first step, using the theoretical filter to process the modulator outputs S_1 , we have:

$$X_1 = V_{pixel} = \widehat{X}_1 + k_1V_{21}(M_1) + \epsilon_1 \quad (4.4)$$

where $k_1 \ll 1$. And after the second step which is for estimating $V_{21}(M_1)$, we have:

$$X_2 = V_{21}(M_1) = \widehat{X}_2 + k_2V_{22}(M_2) + \epsilon_2 \quad (4.5)$$

where $k_2 \ll 1$. Replacing $V_{21}(M_1)$ in equation (4.4) by equation (4.5), finally, the ADC input is equivalent to the sum of the outputs of the theoretical filter plus several error terms:

$$V_{pixel} = \widehat{X}_1 + k_1\widehat{X}_2 + k_1k_2V_{22}(M_2) + \epsilon_1 + k_1\epsilon_2 \quad (4.6)$$

Logically the error in the first step ϵ_1 dominates the conversion error as the impact of ϵ_2 is decreased by k_1 . Normally, we have $\epsilon_1 \gg \epsilon_2 \gg k_1k_2V_{22}(M_2)$. Due to ϵ_1 and ϵ_2 , the theoretical filter (equation (3.18)) is no longer a good estimation of the ADC input. Therefore, a new correction algorithm is required to minimized the conversion error.

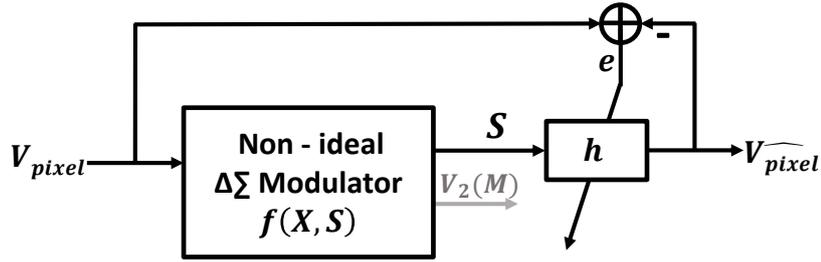


Figure 4.2: Diagram of correction methodology

4.2 Correction methodology

Our purpose is to find an algorithm that reconstructs the ADC input in a linear way regarding modulator output bits or combinations of those. Also, the algorithm should contain an offset correction term. Thus it is equivalent to find a FIR filter plus an offset correction. Namely let denote V_{pixel} the column vector of the input voltages and $V_{\widehat{pixel}}$ the estimators of those input voltages. $V_{\widehat{pixel}}$ is expressed as $S \cdot h^T$ where S is a matrix generated from the bit stream S and h is the line vector of the coefficients of the filter with the offset. The elements of h are then obtained by minimizing the criterion $\|V_{pixel} - S \cdot h^T\|^2$ in a mean square sense as shown in Fig. 4.2. As mentioned in the last chapter, two post-layout simulations were executed respectively in one-step and two-step and all the ADC inputs and the modulator outputs were recorded. All the correction methods are verified on Matlab using these post-layout simulation results.

In order to verify the correction methods, we divided randomly data set into two groups. One group is the learning group which contains 80 percent of the samples in data set while the other group is the testing one which contains the rest of 20 percent samples. The coefficients are estimated using the learning group, then the estimated coefficients are applied directly to the testing group. ER is calculated only using the testing group.

4.3 State-of-the-art correction methods

There exist only few correction methods for $\Sigma\Delta$ ADCs and these correction methods are generally implemented in the digital domain. We first applied them to the post-layout simulation results to see whether we were able to achieve the target resolution.

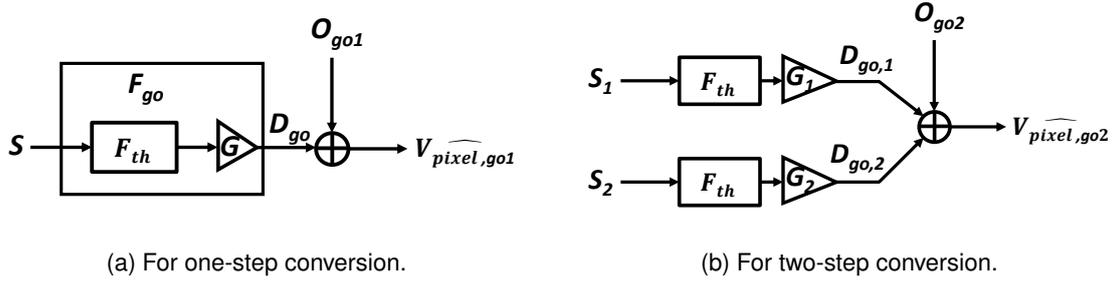


Figure 4.3: Diagram of the correction of gain and offset.

4.3.1 Correction of gain and offset

The simplest correction is to correct gain in conversion errors by multiplying the coefficients of theoretical filter by the same factor and adding a constant to correct the offset. The transfer function of the filter to correct gain is:

$$F_{go}(z) = G(1 + 2z^{-1} + 3z^{-2} + \dots + Mz^{-M+1}) \quad (4.7)$$

where M is the number of bits used for the correction. To be noticed that the input of filter is directly the modulator output bit sequence. Now we explicate how to apply the correction filter for the conversion modes one-step and two-step respectively. For one-step conversion, as shown in Fig. 4.3a, the ADC input can be estimated as:

$$V_{pixel,go1} = D_{go} + O_{go1} \quad (4.8)$$

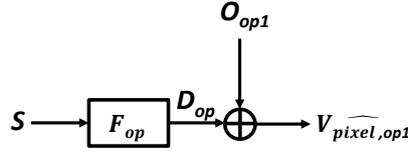
where D_{go} is the output of F_{go} and O_{go1} is the term to correct offset.

For a conversion in two-step mode, as shown in Fig. 4.3b, we use two F_{go} filters with different values of G , and a common offset. The ADC input is finally estimated by:

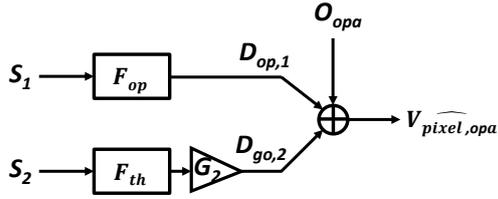
$$V_{pixel,go1} = D_{go,1} + D_{go,2} + O_{go2} \quad (4.9)$$

where $D1_{go}$ and $D2_{go}$ are respectively the output of F_{go} and O_{go2} is to correct offset.

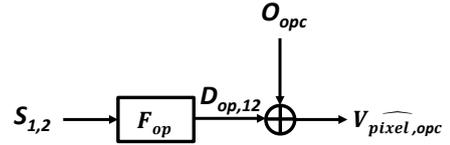
In terms of complexity, by simply correcting gain and offset, only 2 coefficients are required for a one-step conversion while 3 coefficients are required for a two-step one (because the offset is global to both steps).



(a) For a one-step conversion.



(b) For a two-step conversion, in plan A.



(c) For a two-step conversion, in plan B.

Figure 4.4: Diagram of the optimal filter correction.

4.3.2 Optimal filter

An optimal filtering consists in adjusting the coefficients of each bit separately as well as the offset. Thus we can expect a finer correction, with the counterpart of a higher complexity. The transfer function of the optimal filter can be written as:

$$F_{op}(z) = (\alpha_0 + \alpha_1 z^{-1} + \alpha_2 z^{-2} + \dots + \alpha_{1-M} z^{-M+1}) \quad (4.10)$$

where M is the number of clock cycles. Thus for a one-step conversion, as shown in Fig. 4.4a, the ADC input is reconstructed as:

$$\widehat{V_{pixel,op}} = D_{op} + O_{op} \quad (4.11)$$

where D_{op} is the output of F_{op} and O_{op} is the term to correct the offset. In this case, the number of coefficients required increases to $M + 1$.

For a two-step conversion, we can either keep the modulator outputs in each step separate or merge them to a global one, depending on situations. Thus we have two plans to apply the optimal filter. In the plan A, as shown in Fig. 4.4b, the optimal filter is only applied to the modulator outputs of the first step and the theoretical filter with corrected gain is applied to the modulator outputs of the second step. Thus the ADC input is reconstructed by:

$$\widehat{V_{pixel,opa}} = D_{op,1} + D_{go,2} + O_a \quad (4.12)$$

where $D_{op,1}$ is the output of F_{op} , $D_{go,2}$ is the output of F_{go} and O_a is the term to correct offset. In this case, the number of coefficients required is $M_1 + 2$ (M_1 is the number of clock cycles in the first step).

In plan B, as shown in Fig. 4.4c, the modulator outputs of both steps are merged into a single bit sequence. One optimal filter is applied to this new bit sequence. The ADC input is reconstructed by:

$$\widehat{V_{pixel,opb}} = D_{op,12} + O_b \quad (4.13)$$

The plan B is similar to the optimal filter correction for the one-step conversions and $M + 1$ coefficients are necessary, where $M = M_1 + M_2$ is the global number of clock cycles.

4.3.3 Correction results of existing methods

Correction results of the simulation with ideal models

Table 4.1 shows the correction results by employing diverse correction methods to the ideal circuit-level model of our 2^{nd} order $\Sigma\Delta$ ADC. We find that in one-step conversions with $M = 400$, after correcting gain and offset, we obtain $ER = 15.6$ bits, which is nearly to that without correction (15.5 bits), as there is no intrinsic error of gain and offset in an ideal $\Sigma\Delta$ modulator. Fig. 4.5 (a) shows the conversion errors after the correction of gain and offset. When the optimal filter is applied, the ER is improved significantly to 17.2 bits, as the optimal filter can largely minimize the quantization error as shown in Fig. 4.5 (b). To be noticed that the blue points are the samples of group learning while the red points are samples for testing.

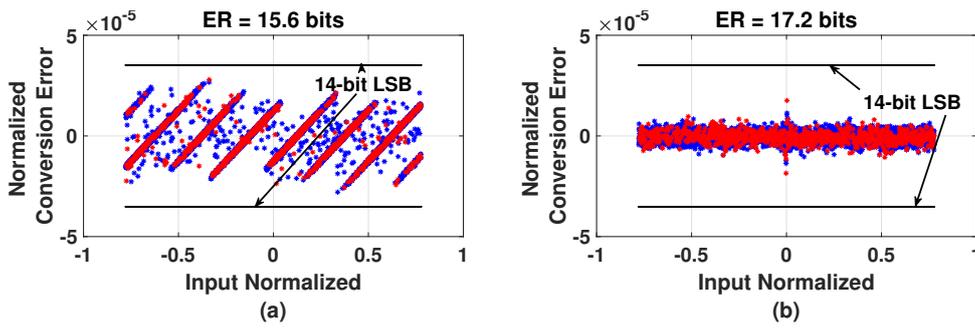


Figure 4.5: Correction results in the one-step mode of an ideal $\Sigma\Delta$ modulator : (a) with the correction of gain and offset; (b) with the optimal filter correction

In two-step conversions, after correcting gain and offset, we obtain $ER = 16.1$, which is the same value as the one without correction. With the optimal filter correction, we observe that if we only correct bit-by-bit for the first step (with plan A), there is no improvement in ER compared to the case without correction. While when we correct bit-by-bit for both second steps (plan B),

	One step	Two steps	
Theoretical filter	15.5	16.1	
Correction of gain and offset	15.6	16.1	
Optimal filter correction	17.2	Plan A	16.1
		Plan B	16.6

Table 4.1: ERs applying the existing correction methods in the simulation of the ideal $\Sigma\Delta$ ADC

we gain the improvement in ER about half a bit. It is due to the fact that the quantization error in a two-step conversion is proportional to the residue ($V_{22}(M)$) of the second step which closely depends on the modulator output of the second step. Comparing the conversion errors in Fig. 4.6 (b) (with the optimal filter in plan B) to those in Fig. 4.6 (a) (with the correction of gain and offset), we find that the conversion errors applying the optimal filter are further minimized. It indicates that for the ideal two-step $\Sigma\Delta$ ADC, the correction of the first step helps only a little to the global resolution.

Correction results of post-layout simulation

The post-layout simulation results in the last chapter are reused to test the correction methods. Table 4.2 shows the summary of correction results in post-layout simulation. Fig. 4.7 (a) shows the form of conversion errors with the correction of gain in offset in the one-step mode. We find that conversion errors are much larger than the target E_{LSB} . In Fig. 4.7 (b), the form of conversion errors seems to be compressed thanks to the optimal filter, and the ER is 13.6 bits, almost achieving our target.

Fig. 4.8a shows the impulse responses of the theoretical filter and the filter of correction gain and offset in one-step mode. It highlights the gain correction that is necessary to have a better ER . Fig. 4.8b shows the impulse responses of the optimal filter in one-step mode. To be noticed

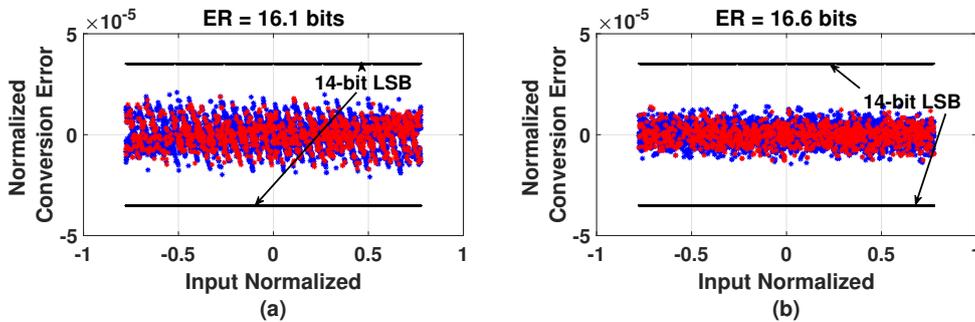


Figure 4.6: Correction results in the two-step mode of an ideal $\Sigma\Delta$ modulator: (a) with the correction of gain and offset; (b) with the optimal filter correction in plan B

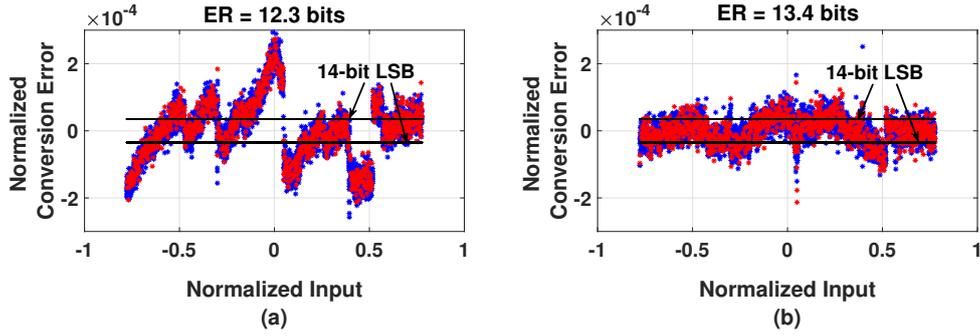
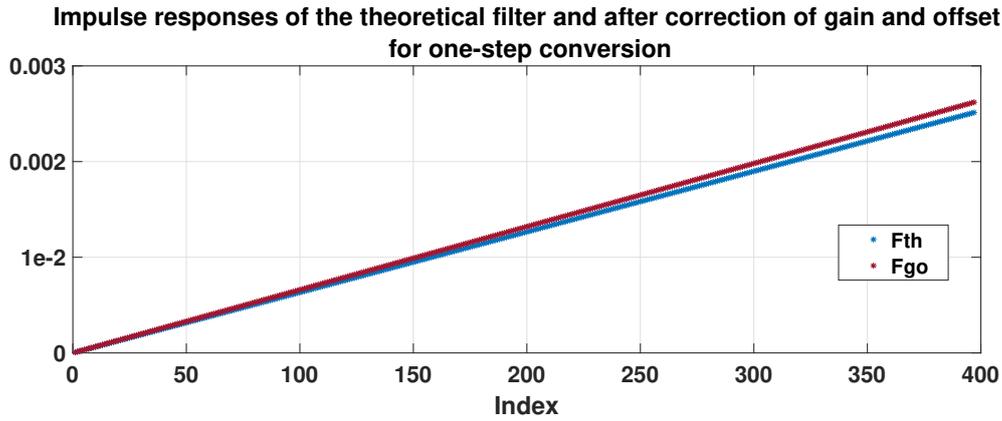


Figure 4.7: Correction results in the one-step mode in post-layout simulation: (a) with the correction of gain and offset; (b) with the optimal filter correction

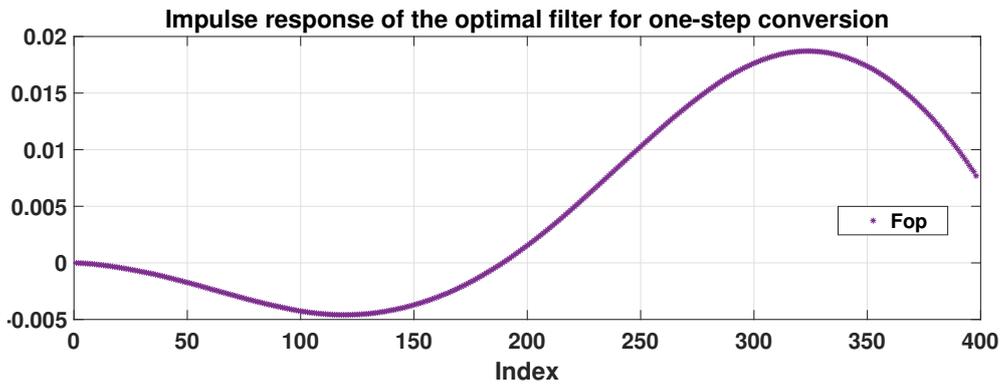
that there several negative coefficients, that is abnormal. It means that these bits are supposed to have negative contribution to the modulator input estimation. This phenomenon will be explicated in the next chapter.

For two-step mode, as shown in Fig. 4.9(a) left, with post-layout results, by only correcting gain and offset, we obtain only $ER = 12.0$ bits. It seems that the second step is limited. It is because that after the first step, ϵ_1 is created and it cannot be estimated by the second step. According to equation (4.6), the contribution of the second step to the final input estimation is multiplied by k_1 , as $k_1 \ll 1$, it is possible that $\epsilon_1 \gg k_1 \widehat{X}_2$. That means the first step conversion error level decides the upper limit of ER . We then apply the optimal filter corrections to the two-step mode respectively with plan A and B. With both plan A and B, the ERs are around 14 bits. Fig. 4.9(b) shows the form of conversion errors using optimal filter correction in plan B. The fact that plan B improves scarcely ER can be explicated as: after the optimal filter correction in plan A, a part of ϵ_1 is compensated while there is residue of ϵ_1 which cannot be completely canceled. And the residue of ϵ_1 is as important as ϵ_2 , or even more important than ϵ_2 . That confirms our previous deduction: ϵ_1 is more critical than ϵ_2 . In other words, if ϵ_1 is not well corrected, the correction of the second step helps little. Fig. 4.10a shows the impulse response of the theoretical filter and that of the filter of correction gain and offset in two-step mode. Compared to the theoretical filter, the values of impulse response of the filter correcting gain and offset for the second step are small. It indicates that the correction of the second step has little contribution to increase the global ER . Fig. 4.10b shows the impulse responses of the optimal filter in two-step mode. Compared to that for one-step conversion, the forms of impulse responses of the optimal filter match better to the theoretical filter.

Table 4.9 shows also the summary of the numbers of coefficients required of the existing correction methods and strengthens the advantage of the two-step mode versus the one-step mode for high resolution.



(a) Impulse responses of the theoretical filter and the filter correcting gain and offset for one-step conversion.



(b) Impulse responses of the optimal filter for one-step conversion.

Figure 4.8: Impulse responses of the filters of existing methods for one-step conversion.

	ER (One step)	Number of coefficients	ER (Two steps)		Number of coefficients
Correction of gain and offset	12.3	2	12.0		3
Optimal filter correction	13.4	401	Plan A	13.9	38
			Plan B	14.0	73

Table 4.2: ERs applying the existing correction methods in post-layout simulation

In conclusion, by applying the existing correction methods, we are almost able to achieve 14 bits in the two-step mode, however in the one-step mode, these techniques are not enough. In the purpose of a characterization of our circuit, we then look for new methods to achieve a *ER*.

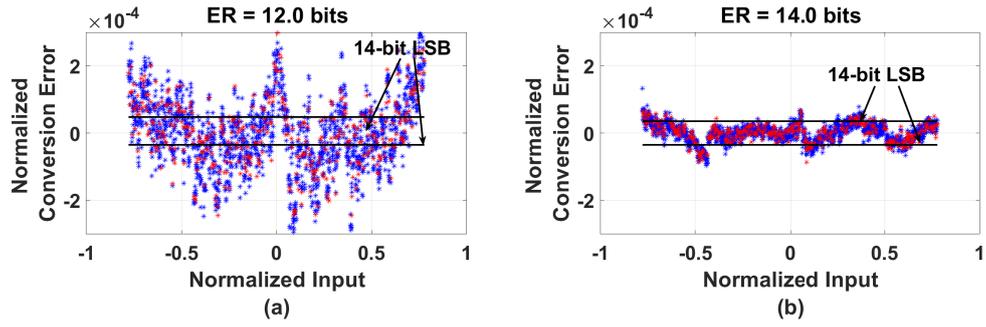
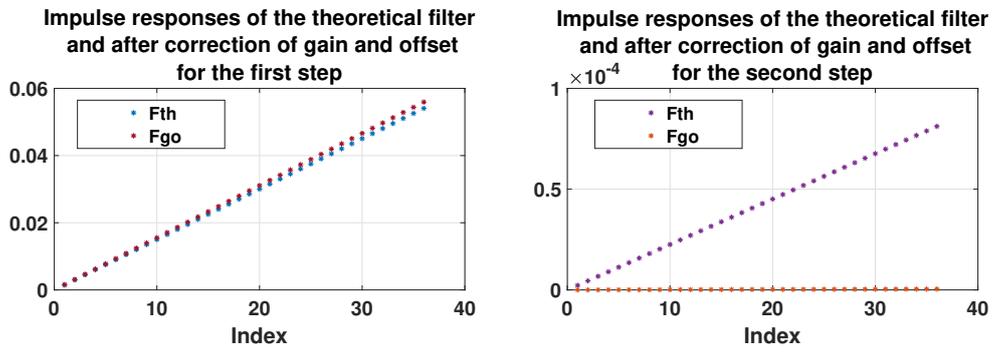
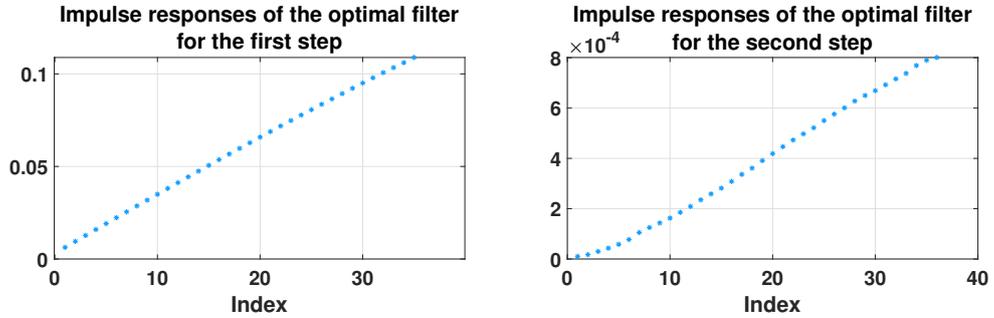


Figure 4.9: Correction results in the two-step mode: (a) with the correction of gain and offset; (b) with the optimal filter correction in plan B.



(a) Impulse responses of the theoretical filter and the filter correcting gain and offset for two-step conversion.



(b) Impulse responses of the optimal filter for two-step conversion.

Figure 4.10: Impulse responses of the filters of existing methods for two-step conversion.

4.4 New proposed correction methods

At this stage, our goal is to find a correction method which is able to make the ADC achieve a higher resolution, without consideration of complexity.

As mentioned, ϵ is a non-linear function of X and S . Despite of the fact that we consider the ADC as a black-box, we look for clues from some internal signals in order to understand

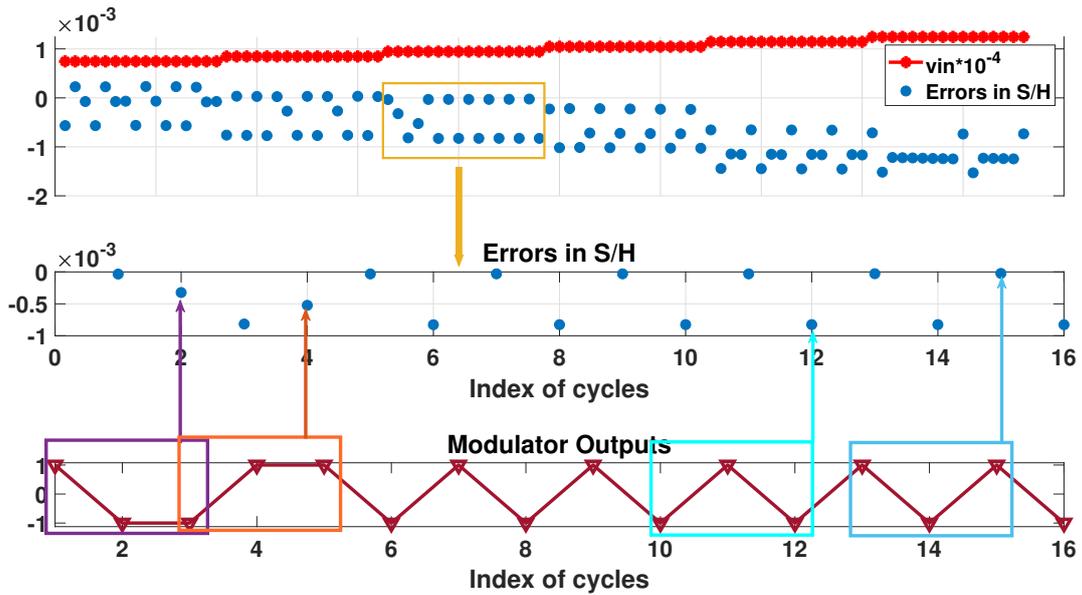


Figure 4.11: Errors in S/H block

the correlation between the conversion errors and the modulator outputs, as they are the only information accessible at the outside of the circuit.

We launched a post-layout simulation in one-step mode with $M = 16$ and recorded the input of S/H vin as well as its output $vmod$ and the modulator outputs at the end of the phase b of the first integrator in each clock cycle. The error of S/H is calculated by $vin - vmod$. We traced the errors of S/H together with the inputs of S/H in Fig. 4.11 and we found that the errors of S/H are not constant. However, since the S/H block only samples vin at the beginning of the conversion, its errors are supposed to be constant. As shown in Fig. 4.11, we discover that the errors between the inputs and the outputs of the S/H block are highly correlated to the current modulator output bit as well as its last two consecutive bits. Therefore, we guess that the conversion errors are also correlated to packages of the modulator output bits. That means not only the value of each modulator output bit but its sequence may make difference to the conversion error as if there was a memory effect. However, the optimal filter correction processes each modulator output bit independently, that may explicate why it is not sufficient. So we should propose a technique taking account of output bit sequence.

4.4.1 Pattern-correcting filter

The proposition of a new algorithm of correction called Pattern-correcting filter is one of my important work. The principle of this algorithm is to detect the combination (pattern) of a pack-

ages of n consequent bits in modulator output bit sequence. In the detection, 2^n new vectors \mathbf{cb}_i , ($i = 1, 2, \dots, 2^n$) of binary bit are created to restore the occurrence of each combination in the package of position m in the modulator output bit sequence. If a certain combination appears in the package of position m , the m th element of the corresponding vector is marked as 1 while the elements in the same position in other vectors are marked as 0. After scanning all packages, the modulator output bit sequence of M bits is expended into 2^n vectors of bit sequence of $M - n + 1$ elements. Fig. 4.12 shows an example of the pattern detection in modulator output bit sequence S for $n = 3$. For each new set of bits, a FIR as equation (4.10) without the offset term is required. And finally, the modulator input is the sum of all the FIR outputs plus a term of offset, as shown in Fig. 4.13. Its expression is:

$$\hat{X}' = \sum_{i=1}^{2^n} (\mathbf{cb}_i \cdot \mathbf{f}_i) + O_3 \quad (4.14)$$

Where \mathbf{f}_i is the vector of coefficients of corresponding FIR.

S	1	-1	-1	1	-1	1	-1	1
CB1 (111)	0	0	0	0				
CB2 (1-11)	0	0	0	1				
CB3 (1-1-1)	1	0	0	0				
CB4 (11-1)	0	0	0	0				
CB5 (-11-1)	0	0	1	0				
CB6 (-111)	0	0	0	0				
CB7 (-1-11)	0	1	0	0				
CB8 (-1-1-1)	0	0	0	0				

Figure 4.12: Illustration of pattern identification

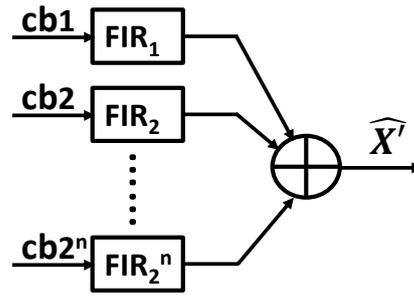


Figure 4.13: Algorithm of Pattern-correcting filter

Analogizing the optimal filter correction for the two-step conversions, we also have plan A,B to apply the pattern-correcting filter correction.

4.4.2 Correction results of pattern-correcting filter techniques applied to post-layout simulation

In order to have a reference, once again, the correction method is first employed to the outputs of the ideal ADC. We apply the pattern-correcting filter with $n = 3$ in one-step conversion as well as in two-step conversion with plan B. The ERs are respectively 17.5 and 16.2. For both conversion modes, the pattern-correcting filter correction has almost the same performance as the optimal filter correction. In other words, there is no need to apply the pattern-correcting filter correction to an ideal $\Sigma\Delta$ ADC, as the optimal filter correction is enough to minimize the quantization error.

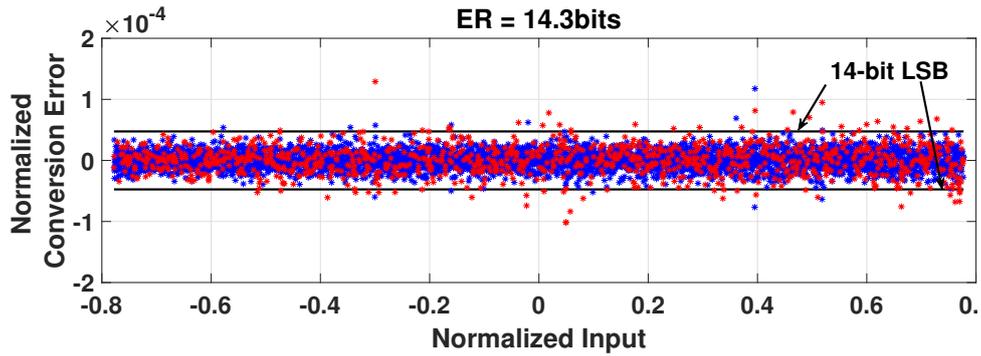


Figure 4.14: Correction result in one-step mode with the pattern-correcting filter correction, $n = 3$

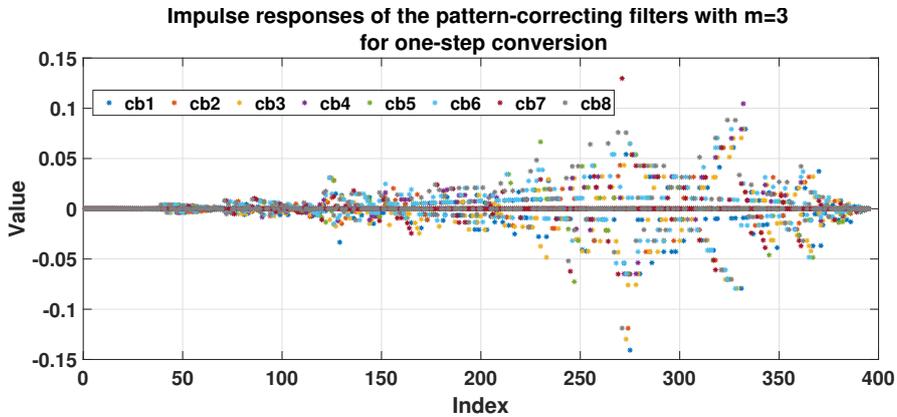


Figure 4.15: Impulse responses of the pattern-correcting filters for one-step conversion.

The correction result of the pattern-correcting filter with $n = 3$ in one-step mode in post-layout simulation is shown in Fig. 4.14. We find that the ER is improved significantly, from 13.4 bits to 14.3 bits. We also try with $n = 2$ and $n = 4$. The correction results of the pattern-correcting filter with different values of n in one-step mode and the number of coefficients are shown in Table 4.3. Compared to the optimal filter correction, generally, the pattern-correcting filter bring about better ER and the best one is when $n = 3$. It is consistent with the simulation result of the errors in the S/H block. Of course, the price to have a higher resolution is the exponentially increasing number of coefficients required.

According to Fig. 4.15 shows the impulse responses of the pattern-correcting filters for one-step conversion. The coefficients in the pattern-correcting filters are irregularly and most of them are 0 because of the absence of the pattern at certain position. Fortunately, in two-step mode, there are less bits for the correction. Since it has been illustrated in the last section that ϵ_1 is more critical than ϵ_2 , at this time, we only employ the pattern-correcting filter with plan A and B. The correction results of the pattern-correcting filter with different values of n in two-step mode and

Configuration	ER (bits)	Number of coefficients
$n = 2$	13.9	1596
$n = 3$	14.3	3184
$n = 4$	14.0	6352

Table 4.3: Comparison of the pattern-correcting filters in one-step mode based on post-layout simulation

Configuration	ER (bits)	Number of coefficients	Figure of Merit (FM)	
$n = 2$	plan A	14.5	142	0.0035
	plan B	14.6	285	0.002
$n = 3$	plan A	15.0	274	0.0036
	plan B	15.2	561	0.0021
$n = 4$	plan A	15.1	530	0.0020
	plan B	15.4	1105	0.0012
$n = 5$	plan A	15.2	1026	0.0012
	plan B	15.6	2177	0.00073

Table 4.4: Comparison of the pattern-correcting filters in two-step mode based on post-layout simulation

the number of coefficients are shown in Table 4.4. It is possible to have a resolution over 15 bits by applying the pattern-correcting filter corrections. Generally, the more coefficients we employ, the better resolution we obtain. In addition, the larger the pattern size n , the more improvement in plan B compared to plan A, because ϵ_1 is better corrected with large n . However, there is a special case of plan A with $n = 3$, in this case, with less coefficients, we gain higher ER compared to the case of plan B with $n = 2$.

We define a figure of Merit $FM = (ER - 14)/\text{Number of coefficients}$ so that to quantify the price paid for the increase of each bit, from 14 bits. As shown in Table 4.4, considering the resolution and the complexity of the correction, $n = 3$ in plan A is a choice with good cost-effective ratio.

Fig. 4.16 shows the impulse responses of the pattern-correcting filters for two-step conversion. In general, it is similar to that in one-step mode.

The ADC can achieve higher ER , until 15.6 bits, applying the pattern-correcting filter correction. In fact, the pattern-correcting filters only provide a theoretical possibility of ER as, they need significant numbers of coefficients. That is hard to realize in reality.

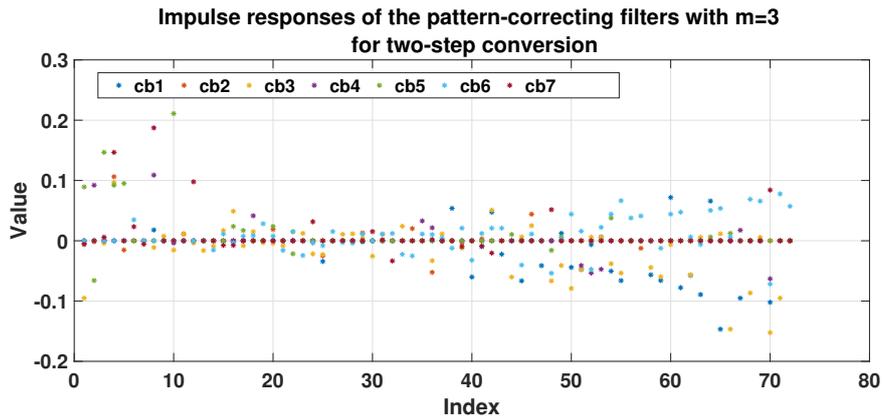


Figure 4.16: Impulse responses of the pattern-correcting filters for two-step conversion.

4.5 Corrections results for multi-step conversions

We just found that, in two-step mode with $M = 72$, after the optimal filter correction in plan B, the ER is around 14 bits. According to the theory of the $\Sigma\Delta$ ADC, if we increase the global number of clock cycles M , we suppose to obtain a high resolution. However, due to the constraint of conversion speed, $M = 72$ is almost the limit of the global number of clock cycles when applying a 20 MHz system clock. So we cannot increase M largely. While the slight increasing of M cannot bring a significant improvement of ER . Thus it is nature that we think about executing the conversions in three even four steps which are able to achieves higher ER with the same value of M .

Another four post-layout simulations with different configurations have been done both for the three-step and four-step conversions with the same simulation conditions. The simulations in multi-step conversions are configured in infinite mode, respectively with $M_1 = 8$ and $M_1 = 16$, keeping global number of clock cycles $M = 72$.

We correct firstly the gain of each step as well as the offset to see whether we can obtain a better ER with increase of very few coefficients. The ERs after correction of gain and offset in each configuration are shown in Table 4.5. We can find that with the correction of gain and offset, ERs correlate to M_1 . Apart from it, all ERs of multi steps after the correction of gain and offset are much worse than that of two steps, as M_1 in multi-step mode is little than that in two-step mode. These phenomena prove once again that the final correction results mainly depend on the first step.

Then we apply respectively the optimal filter and the pattern-correcting filter in plan B to the modulator outputs in multi-step conversions, in order to see whether it is possible to achieve a

Configuration	ER (bits) of theoretical model	ER (bits) after correcting gain and offset	Number of coefficients
3 steps, $M_1 = 8$	15.3	9.2	4
3 steps, $M_1 = 16$	19.1	10.8	4
4 steps, $M_1 = 8$	19.0	9.2	5
4 steps, $M_1 = 16$	19.3	10.8	5

Table 4.5: ERs after correction of gain and offset in multi-step modes.

Configurations	three-step mode		four-step mode	
	$M_1 = 8$	$M_1 = 16$	$M_1 = 8$	$M_1 = 16$
ER of theoretical model	15.9	19.2	19.1	19.5
ER in post-layout simulation	11.6	12.8	11.7	12.8
Numbers of coefficients	73			

Table 4.6: ERs corrected with the optimal filter in plan B of multi-step modes.

Filter configuration, numbers of coefficients	Type of model	three-step mode		four-step mode	
		$M_1 = 8$	$M_1 = 16$	$M_1 = 8$	$M_1 = 16$
$n = 2, 285$	Theoretical	15.9	19.3	18.9	19.7
	Post-layout	12.2	13.4	12.3	13.4
$n = 3, 561$	Theoretical	16.0	19.6	19.5	19.7
	Post-layout	13.2	14.1	13.2	14.1
$n = 4, 1105$	Theoretical	16.1	19.6	19.5	19.6
	Post-layout	13.4	14.2	13.3	14.2
$n = 5, 2177$	Theoretical	16.1	19.6	19.6	19.6
	Post-layout	13.5	14.2	13.4	14.2

Table 4.7: ERs corrected with the pattern-correcting filter in plan B of multi-step modes.

higher ER with more complex corrections. As shown in Table 4.6 and Table 4.7, ERs of multi-step conversions are not better than that of two-step conversion and ER also strongly depends on the accuracy of the first step. The degradation in multi-step mode may be caused by the inter-stage errors. According to the correction results, multi-step conversions do not help to obtain a higher resolution.

4.6 Correction method simplification

The case around or achieving 14 bits with least number of correction coefficients is the conversion in two steps with the optimal filter correction in plan A, which requires 38 coefficients. But in our context, because we handle column-parallel ADCs for an HD image sensor, the total number of coefficients may be overwhelming. In order to reduce the number of coefficients while targeting

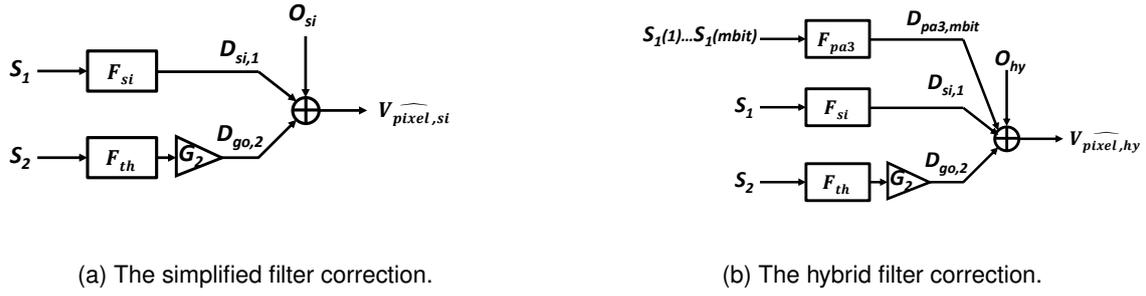


Figure 4.17: Diagram of correction methods for two-step conversion after simplification

14-bit, we propose to simplify the correction methods. Taking the case of a conversion over two steps applying the optimal filter correction in plan A which requires 38 coefficients as a starting point, we try to simplify the correction methods as possible and keep the ADC resolution stay over 14 bits in two-step mode.

4.6.1 Simplified Filter

Compared to a FIR, an IIR filter requires less coefficients. For this reason, we attempt to find an IIR whose impulse response approaches that of the optimal filter over M inputs. The transfer function of that filter is:

$$F_{si}(z) = \frac{p_0 + p_1 z^{-1} + \dots + p_n z^{-n}}{q_0 + q_1 z^{-1} + \dots + q_m z^{-m}} \quad (4.15)$$

To do that, we can use Steiglitz-McBride iteration which allows finding an IIR filter with a prescribed time-domain impulse response by knowing the expected impulse response and the number of zeros and the number of poles.

Thus in two-step mode, as shown in Fig. 4.17a, using the simplified optimal filter correction with plan A, the ADC input can be estimated as:

$$\widehat{V_{pixel,si}} = D_{si,1} + D_{go,2} + O_{si} \quad (4.16)$$

where $d_{si,1}$ is the output of F_{si} for the first step, $d_{go,2}$ is the output of F_{go} for the second step and O_{si} is to correct offset.

With the help of the Steiglitz-McBride algorithm, an IIR with 2 poles and 2 zeros is found to approximate the 36th-order optimal filter in the last section which corrects the first step. Since it has only 2 poles, there is no risk of instability of the IIR. The comparison of correction results with the optimal filter and the simplified optimal filter in plan A in post-layout simulation are shown in Fig. 4.18. The ER after applying the simplified optimal filter correction is close to 13.1 bits.

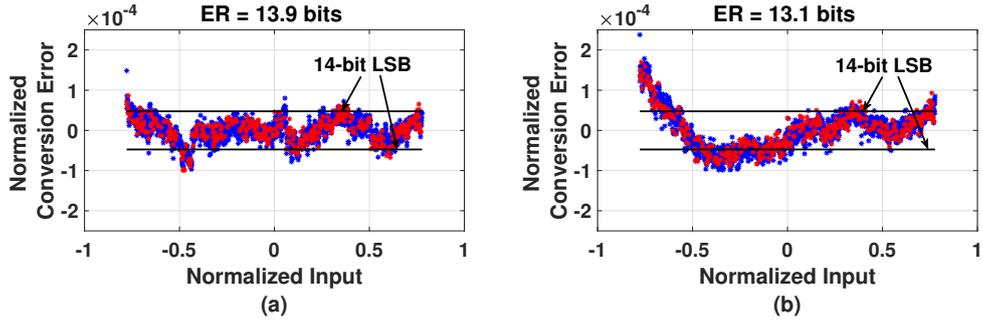


Figure 4.18: Comparison of correction results in the two-step mode: (a) with the optimal filter correction in plan A; (b) with the simplified optimal filter correction in plan A

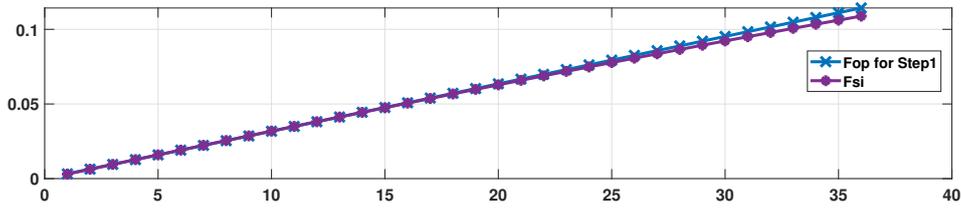


Figure 4.19: Comparison of the impulse responses in the optimal filter and the simplified filter.

However, it the simplified filter reduces number of coefficients to 8 bits.

Fig. 4.19 shows the impulse response of the optimal filter and the impulse response of the simplified filter with 36 impulse inputs. According to Fig. 4.19, the coefficients for MSBs are relatively poorly estimated. That is the reason of the degradation in ER applying the simplified filter.

We also tried to increase the order of IIR to see if we could get a higher ER . For 3 poles and 3 zeros, ER is still 13.1 bits while for 4 poles and 4 zeros, ER slightly increase to 13.3 bits. However, when numbers of poles and zeros increase to 5, ER degrades significantly to 11.6 bits. Considering stability, we chose 2 poles and 2 zeros in priority.

4.6.2 Hybrid filter

Since the simplified optimal filter is an approximation of an optimal filter, degradation in ER is expected. In order to compensate the loss of resolution, a pattern-correcting filter of 3-bit package ($n = 3$) is employed to correct the combinations of the first m MSBs of the modulator output in the first step. As shown in Fig. 4.17b, combining the simplified optimal filter correction and the pattern-correcting filter correction, the corrected ADC input is estimated as:

$$\widehat{V_{pixel,hy}} = D_{si,1} + D_{pa3,m\text{bit}} + D_{go,2} + O_{hy} \quad (4.17)$$

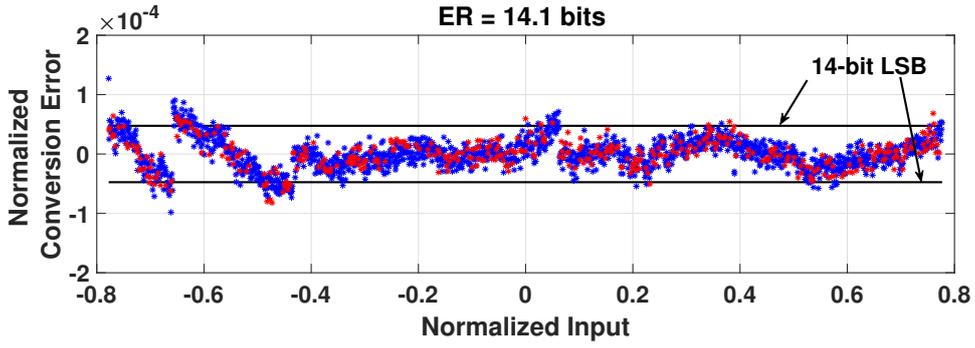


Figure 4.20: Correction result in the two-step mode with the hybrid filter correction in plan A

where $d_{si,1}$ is the output of simplified optimal filter F_{si} for the first step, and $D_{pa3,m\text{bit}} = \sum_{i=1}^8 \text{cbi} \cdot \text{fi}$ which is the sum of the outputs of corresponding FIR_i , $d_{go,2}$ is the output of F_{go} for the second step and O_{hy} is the term to correct offset.

The correction results of hybrid filter with $n = 3$ in post-layout simulation are shown in Fig. 4.20. In our application, $m = 3$ is enough to achieve the desired resolution and the number of coefficients necessary is 16.

We also tried to increase the order of IIR (the simplified filter), however, for the order varying from 2 to 4, ER stays at 14.1 bits and for an IIR of 5th order, ER is only 12.3 bits. Combine previous results of the simplified filter, we can conjecture that an IIR with 4 poles and 4 zeros could better approximate the MSB of the impulse response of the FIR. That is why the ER of the simplified filter increases. However, since the hybrid filter corrects the first 3 MSB supplementary according their pattern, the advantage of the IIR of fourth order becomes no longer notable. Considering the complexity and the stability, we chose an IIR of second order as the simplified filter.

4.6.3 Simplified hybrid correction

In fact, due to circuit characteristics, not all patterns of a given number of bits appear at the same frequency. For this case of 3 bits, occurrences of patterns $\{-1-11\}$, $\{-11-1\}$, $\{1-1-1\}$, $\{1-11\}$ are more frequent than the others. If we only take into account these 4 combinations rather than 8, the pattern-correcting filter correction can be simplified and its number of coefficients will be cut in half. The correction results of simplified hybrid filter with $m = 3$ in post-layout simulation are shown in Fig. 4.21. The result seems to be exactly the same as that of the hybrid filter correction. However, the number of coefficients required is further decreased to 12.

Fig. 4.22 shows the comparison of performances of the hybrid correction and the simplified

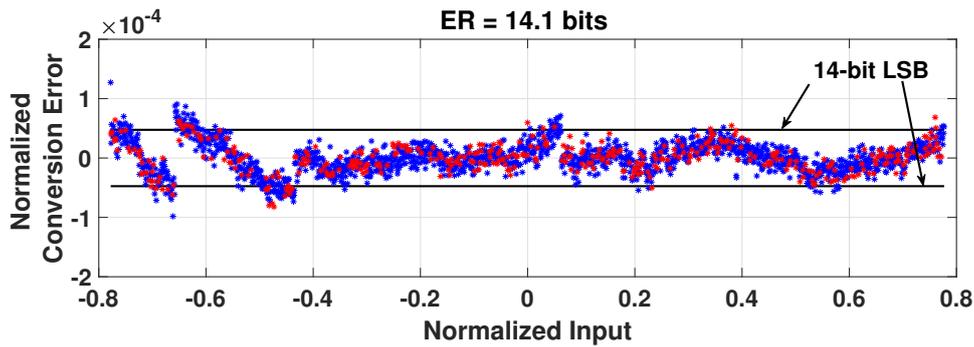


Figure 4.21: Correction result in the two-step mode with the simplified hybrid filter correction in plan A

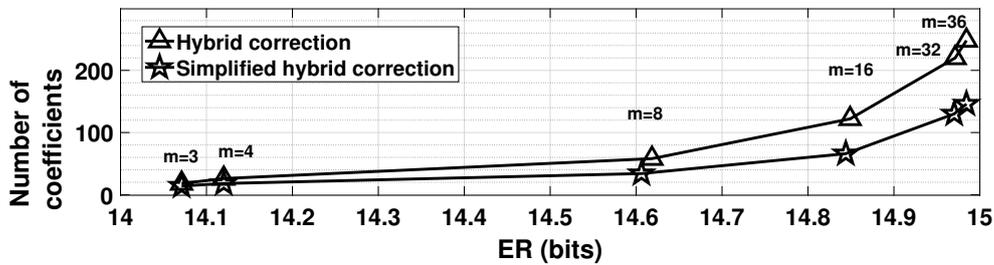


Figure 4.22: Comparison of the simplified hybrid filter and the hybrid filter in plan A with different m

hybrid correction with different value of m in pattern-correcting filter. For $m = 3$ and $m = 4$, there is no degradation in ER owing to the absence of other combinations. For other values of m , the difference in ER are also very small while we gain more in complexity with the increase of m .

4.7 Coefficient quantization

It should be noticed that not only the number of coefficients, but also the number of bits to encode these coefficients has impacts on complexity. After a parametric simulation, the minimum numbers of bits to encode a coefficient in different types of filters are found, under the condition that the degradation of ER is lower than 0.1 bit. Table 4.8 shows the number of bits to memorize a coefficient in a FIR filter while Table 4.9 shows the numbers of bits respectively to memorize a coefficient of numerator and of denominator in IIR filter. Table 4.10 shows the total numbers of bits to memorize all coefficients as well as the performance of some correction methods. In brief, the simplified hybrid correction cost less compared to optimal filter correction for an equal ER of

Correction method	number of coding-bits
Gain and offset	14
Optimal filter	18
Simplified optimal filter (O_{si})	18
Hybrid (k_{alt} and O_{hy})	20

Table 4.8: Numbers of bits to code a coefficient in the FIRs of different corrections

position	number of coding-bits
Numerator	16
Denominator	18

Table 4.9: Numbers of bits to code a coefficient in simplified optimal filter (type IIR)

Correction method	Number of coefficients, number of encoding bits	ER (bits)
Gain and offset	3; 42	12.0
Optimal filter in plan A	38; 684	14.0
Simplified optimal filter in plan A	8; 142	13.1
hybrid filter in plan A, $m = 3$	16; 306	14.1
Simplified hybrid filter in plan A, $m = 3$	12; 222	14.1

Table 4.10: Comparison of some correction methods in two-step mode with quantized coefficients
14-bits.

The comparison of corrections methods in two-step mode is shown in Fig. 4.23. Considering the resolution and the complexity, we find a best trade-off correction method: the simplified hybrid filter in plan A with $m = 3$. With this correction, we finally obtain 14 bits only with 12 coefficients and 222 encoding bits.

4.8 Resolution optimization for two-step conversion

According to previous analysis, as ϵ_1 is the main contributor to the conversion error, improving the resolution of the first step is more effective to global resolution improvement. Since increasing M_1 provides more information to the correction of step 1 while the number of coefficients required for the corrections of simplified optimal filter or optimal filter with plan B will not increase, the ER may be optimized without additional complexity. Fig. 4.24 shows the global ER in the function of M_1 corrected by a 72-order optimal filter in post-layout simulation, (the coefficients are non-quantized). When M_1 is increased to 48, ER is improved while if M_1 is further increased to 60, ER is degraded compared to $M_1 = 36$. This is because as $M_1 = 60$, regardless of the well-corrected first step,

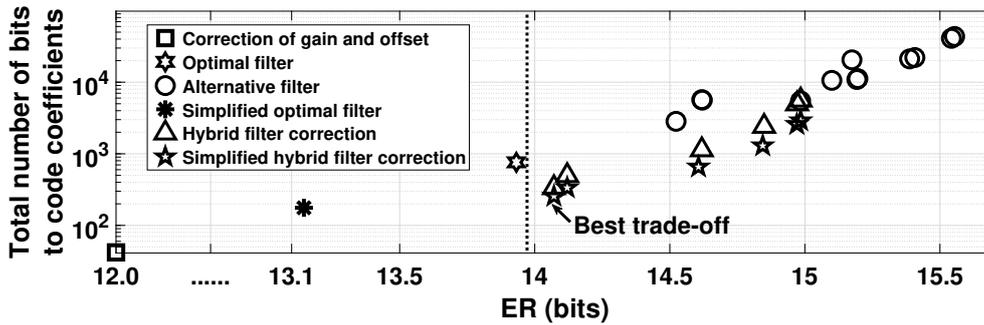


Figure 4.23: Summary of correction methods.

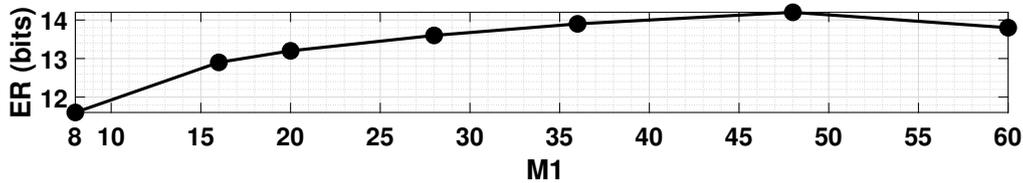


Figure 4.24: With the optimal filter correction in plan B, global equivalent resolution in two-step mode in the function of M_1

Correction methods m	ER (bits)		Number of bits to code
	$M_1 = 36$	$M_1 = 48$	
Simplified optimal	13.2	13.7	142
Simplified hybrid, $m = 3$	14.1	14.2	222
Simplified hybrid, $m = 4$	14.1	14.3	302
Simplified hybrid, $m = 8$	14.6	14.6	622

Table 4.11: Comparison of ER when $M_1 = 36$ and $M_1 = 48$, using simplified hybrid correction in plan A.

the theoretical global resolution is too low. Table 4.11 compares the results of some correction methods with coefficients quantized in the cases where $M_1 = 36$ and $M_1 = 48$. With the same complexity, latter has better ER . Among all situations, the best trade-off is $ER = 14.2$ bits costing 222 bits when $M_1 = 48$, of which conversion error is shown in Fig. 4.25.

4.9 Conclusion

In this chapter, we first applied the existing correction methods to our ADC. The post-layout simulation results show that the existing correction methods are not sufficient to make our ADC achieve

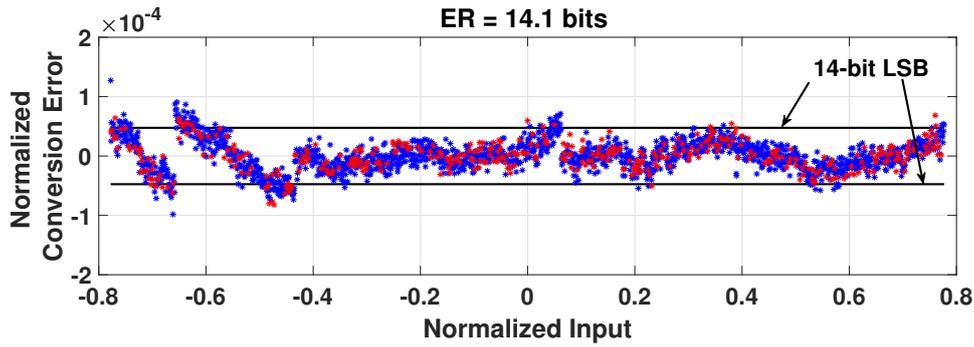


Figure 4.25: Conversion error when $M_1 = 48$ using simplified hybrid correction in plan A with $n = 3, m = 3$

the desired resolution. Therefore, we propose a new correction method: the pattern-correcting filter correction. It can be an alternative method to reconstruct the input of an $\Sigma\Delta$ ADC. Although it is more complex, it brings about higher resolution.

However, for column-parallel ADCs, the constraint of complexity is critical. In order to make the correction method implementable in a real circuit, we propose the simplified versions of the existing correction method and the new-proposed pattern-correcting filter correction. To guarantee the target resolution, we combine these simplified correction methods together and find a best trade-off correction plan. In this way, we obtain the 14-bit target ADC resolution taking 12 coefficients. This complexity is acceptable for column-parallel ADCs.

Among the correction methods which are able to make ER achieve 14 bits, the complexity of the best trade-off correction is the lowest. However, it is still possible to obtain a higher resolution keeping the same complexity of correction method by optimizing the number of clock cycles of the conversion in the first step. Post-layout simulation shows that with $M_1 = 48$, after the best trade-off correction, ER has its maximum value.

To be noticed that all the post-layout simulations have been done at $25^\circ C$, more research about the impacts of temperature over the correction performance are to be carried out.

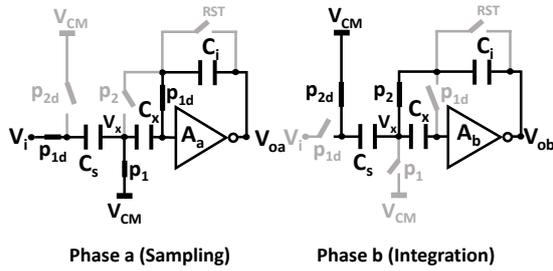
Chapter 5

Circuit defects modeling

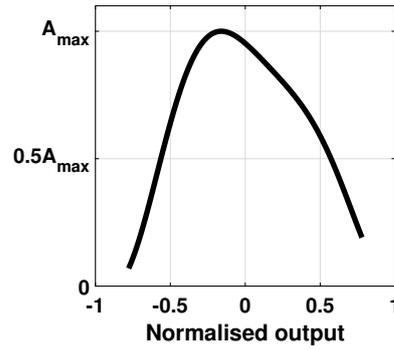
In the last chapter, we investigated the correction methods for our $\Sigma\Delta$ ADC and proposed a method which provides the best trade-off between resolution and complexity, treating the ADC as a black-box. However, we still want to figure out the sources causing the degradation of ADC. So the goal of this research is to find a model of our ADC which can explicate the degradation and the ADC's behavior observed in post-layout simulation.

The integrator is an important component of the low-pass $\Sigma\Delta$ ADCs as its errors created in each clock cycle participate into the conversion in the following loops. In our case, the integrators may bring about much more errors because of the inverter-based amplifier. One of its well-known issue is its non-linear limited amplifier gain. There are also other classical non-idealities, such as capacitor mismatch and signal-dependent capacitance. These circuits defects change the behavior of the integrator by shifting poles and zeros in the integrator transfer function and then change the transfer function of the $\Sigma\Delta$ modulator. In terms of the static characteristics, shift of poles and zeros in the transfer function may lead to the variation of DNL and INL, because of the mismatch between the transfer functions of the $\Sigma\Delta$ modulator and the digital reconstruction filter. While in terms of the dynamic characteristics, it may cause the change of the frequency characteristics which makes difference to SNDR or SNR. In our case, since the ADC inputs are DC signals, we only take into account its static characteristic INL (ER) as the criteria to judge the ADC's performance. As illustrated in Chapter 3, ER is affected by the change of the modulator's transfer function.

In post-layout simulation, we observed that some intern signals in integrators vary abnormally. However, the model with the classical circuit non-idealities such as non-linear amplifier gain or capacitor mismatch cannot explicate the variation of these signals. For this reason, we have an assumption: the schematic of integrators is changed due to the parasitic capacitors. Therefore,



(a) Schematic of inverter-based SC integrator with non-linear amplifier gain.



(b) Amplifier gain in function of its output

according to the clues in post-layout simulation, we propose a new integrator model with some important parasitic capacitors. With this new model, the variation of some intern signals can be explicated.

Once the new model is proposed, how to validate it becomes a question. Simply comparing the ER seems to be too general. As we cannot validate the proposed model directly, we compare not only the value of ER but also the form of conversion errors in several scenes based on the results of post-layout simulation. We compared the forms of conversion errors after different corrections of the proposed error model to those in post-layout simulations.

After validating the proposed model, more studies about the correction methods as well as modulator characteristics has been done based on the new integrator model. This knowledge is useful to enhance future designs.

5.1 Classical circuit non-idealities

5.1.1 Non-linear amplifier gain

The non-linear amplifier gain is one of the most common issue of $\Sigma\Delta$ modulator as it is the main non-linear error source of an SC integrator as well as a $\Sigma\Delta$ ADC. The impact of the finite amplifier gain in SC integrators has been analyzed in detail by many papers: [93][94][44], researches show that the finite amplifier gain causes degradation of the frequency characteristics. [43] and [95] point out that the effect of integrators finite gain on a $\Sigma\Delta$ modulator is the leakage in integration and output $SNDR$ remains almost unchanged for low-frequency input if the amplifier gain is great enough ($> 60dB$).

We first modeled this defect into our circuit. Let's look at back to Fig. 5.1a and define $A_a(n)$ and $A_b(n)$ respectively as the time-varying gain in phase a and phase b at clock n . Therefore the

amplifier inputs become respectively $\frac{V_{o,a}(n)}{A_a(n)}$ and $\frac{V_{o,b}(n)}{A_b(n)}$. Then according the fact that from phase a to phase b, we always have: $\Delta Q_s = -(\Delta Q_x + \Delta Q_i)$, the relationship between $V_{o,b}(n)$ and $V_i(n)$, $V_{o,a}(n)$ can be expressed as:

$$V_{o,b}(n) = k_{ab1}(n)V_{o,a}(n) + k_{ib1}(n)V_i(n) - k_{ob1}(n)V_{of} \quad (5.1)$$

with:

$$k_{ab1}(n) = \frac{C_i + (2C_i + C_s)\frac{1}{A_a(n)}}{C_i + (C_i + C_s)\frac{1}{A_b(n)}} \quad (5.2)$$

$$k_{ib1}(n) = \frac{C_s}{C_i + (C_i + C_s)\frac{1}{A_b(n)}} \quad (5.3)$$

$$k_{ob1}(n) = \frac{(1 + \frac{1}{A_a(n)})C_i + (C_s + C_i)(\frac{1}{A_b(n)} - \frac{1}{A_a(n)})}{C_i + (C_i + C_s)\frac{1}{A_a(n)}} \quad (5.4)$$

And from phase b at clock cycle $(n - 1)$ to phase a at clock n , we have $\Delta Q_x = -\Delta Q_i$, thus $V_{o,a}(n)$ can be expressed as:

$$V_{o,a}(n) = k_{aa1}(n)V_{o,a}(n-1) + k_{ia1}(n)V_{o,b}(n-1) - k_{oa1}(n)V_{of} \quad (5.5)$$

with:

$$k_{aa1}(n) = -\frac{(C_i + C_s)\frac{1}{A_a(n)}}{C_i + (C_i + C_x)\frac{1}{A_a(n)}} \quad (5.6)$$

$$k_{ba1}(n) = \frac{C_i + (C_i + 2C_x)\frac{1}{A_b(n-1)}}{C_i + (C_i + C_x)\frac{1}{A_a(n)}} \quad (5.7)$$

$$k_{oa1}(n) = \frac{C_i(1 + \frac{1}{A_a(n)}) + C_x(\frac{1}{A_a(n)} - \frac{1}{A_b(n-1)}) - (C_i + C_x)(\frac{1}{A_a(n-1)} - \frac{1}{A_b(n-1)})}{C_i + (C_i + C_x)\frac{1}{A_a(n)}} \quad (5.8)$$

Replacing $V_{o,a}(n)$ in equation (5.1) by equation (5.5), we obtain:

$$V_{o,b}(n) = k_{ib1}(n)V_i(n) + k_{ab1}(n)k_{ba1}(n)V_{o,b}(n-1) + (k_{ab1}(n)k_{ba1}(n) - k_{ob1}(n))V_{of} + k_{ab1}(n)k_{aa1}(n)V_{o,a}(n-1) \quad (5.9)$$

If gain is infinite, equation (5.9) returns to equation (3.22). However, due to this non-linear gain, the term $V_{o,a}(i)$ ($i = 1, 2, \dots, n - 1$) as well as offset can not be eliminated completely. Besides,

coefficients of $V_{o,b}(n-1)$ and $V_i(n)$ in equation (5.9) are changed compared to equation (3.22) which can be considered as shift of poles and zeroes in the z -domain transfer function.

Non-linear gain brings about some error terms in both the numerator and denominator in equations (5.2), (5.3), (5.4) and (5.6), (5.7) (5.8), in the form of reciprocal of gain or difference of reciprocal of gain. For the non-linear amplifier model, the gain of the amplifier is modeled in a non-linear way [75]:

$$A(V_o) = \frac{A_{max}}{(p_0 + p_1 V_o + p_2 V_o^2 + \dots + p_8 V_o^8)} \quad (5.10)$$

where $p_x (x = 0, 1, \dots, 8)$ are carefully chosen to make the $A(V_o)$ vary between 0 and A_{max} . Fig. 5.1b shows the amplifier gain $A(V_o)$ as a function of integrator output V_o .

5.1.2 Capacitor mismatch

For an ideal situation, we chose $C_x = C_i$ and $C_s = aC_i$ (a is the coefficient of V_i in equation (3.22)). However, due to fabrication technology, capacitor mismatch cannot be avoid. Thus, in reality, we have $C_x = (1 + \alpha_x)C_i$ and $C_s = a(1 + \alpha_s)C_i$ where α_x and α_s are mismatch ratio. On one hand, the main impact of the mismatch is the change of the coefficient of V_i ; while in another hand, there is coupling of capacitor mismatch to the errors brought by non-linear amplifier gain.

5.1.3 Simulation of the modulator model with classic defects

Based on equation (5.9), a high-level model of inverter-based SC integrators with classic defects as well as a model of a 2^{nd} $I\Sigma\Delta$ modulator composed of those integrators have been built on Matlab. Several simulations of the $I\Sigma\Delta$ modulator model with classic defects in one-step with $M = 400$ have been done by sweeping parameters. Also by observing directly the signals in post-layout simulation, we found that there was an offset ($V_{of} = -0.027V$) in both integrators; and there was also an offset of threshold voltage in the comparator ($V_{th} = 0.021V$). For this reason, we added these imperfections into the Matlab model.

Fig. 5.2 shows the ERs in the function of A_{max} in the simulations. The ER of an ideal modulator after correcting gain and offset (in fact there is no intrinsic gain and offset of an ideal modulator) and after the optimal filter correction are respectively 15.5 bits and 17.5 bits. From Fig. 5.2, we find that the modulator resolution is not sensitive to the amplifier gain as when $A_{max} > 80dB$, the degradation in ER can be ignored. The same discovery is also valid for capacitor mismatch. Fig. 5.3(a) shows the ERs in the function of α_s while Fig. 5.3(b) shows ERs in the function of α_x in the simulations. For mismatch of both C_s and C_x , the ERs with correction of gain and offset

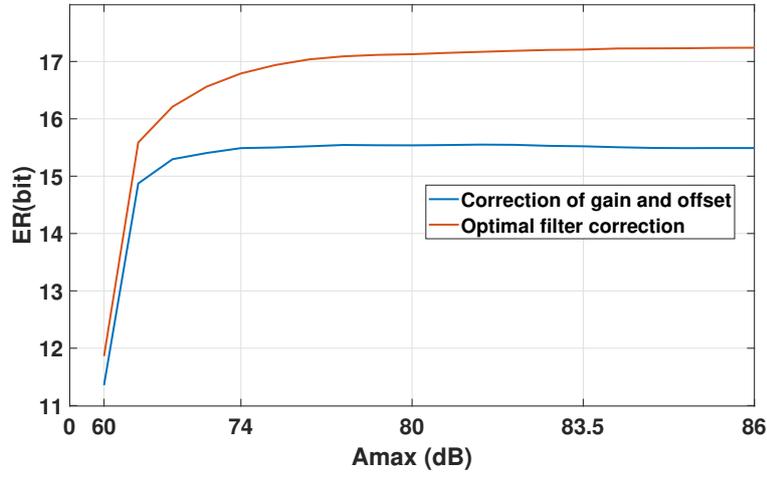


Figure 5.2: ER in the function of A_{max}

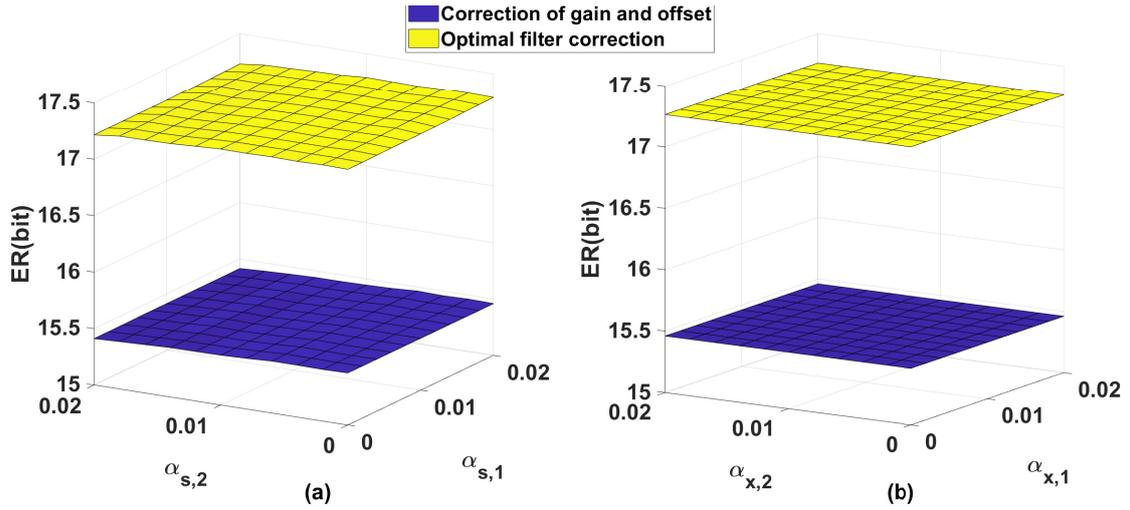


Figure 5.3: ER in the function of capacitor mismatch ratios: (a) C_s/C_i ; (b) C_x/C_i

are around 15.4 bits and the ERs after the optimal filter correction are around 17.3 bits. In fact, the mismatch of C_s and C_x lead to an offset and a gain in conversion errors compared to the ideal modulator.

According to the previous results, we can conclude that the main reasons of the degradation in our $I\Sigma\Delta$ modulator are the not finite non-linear amplifier gain and capacitor mismatch.

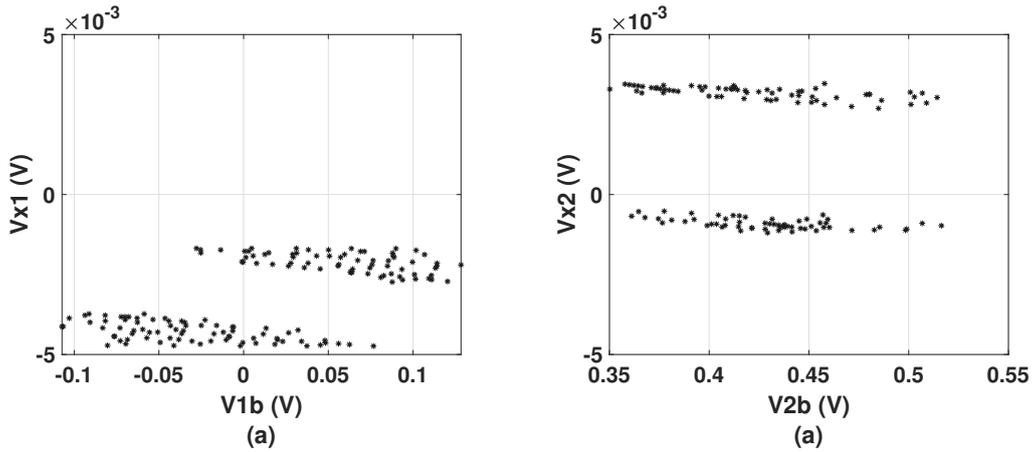


Figure 5.4: $V_{x,b}$ in the function of the integrator outputs $V_{o,b}$

5.2 New inverter-based SC integrator model with parasitic capacitors

5.2.1 Abnormal Variation of Certain Intern Signals in Integrators

As illustrated in Chapter 3.2.1, for an ideal SC integrator, $V_{x,b}$ is supposed to be a constant value ($V_{x,b} = V_{cm}$). Also even with the previous classic circuit non-idealities, $V_{x,b}$ will still remain constant. Fig. 5.4 shows values of $V_{x,b}$ of two integrators in the first eight clock cycles. However, from Fig. 5.4, we find that in both integrators, $V_{x,b}$ varies with an amplitude upon to 5 mV in post-layout simulation. This value is too large to the normal value caused by the limit of simulation accuracy. So we think the classic circuit non-idealities are not enough to model our integrator, there must be other defects.

5.2.2 Discovery of parasitic capacitors

When we plot $V_{x,b}$ in the function of $V_{o,b} - V_{o,a}$, we obtain some interesting discovery. From Fig. 5.5, we found that in both integrators, when the value of S is fixed, $V_{x,b}$ is proportional to the difference of the integrator outputs between two phases. Since there is tight correlation between $V_{x,b}$ and the integrator output, we consider that there is a parasitic capacitor (C_p in Fig. 5.6) between the amplifier input and output. This capacitor exists systematically in an inverter-based amplifier, whose schematic is shown in Fig. 3.8, due to the parasitic capacitors in some transistors, such C_{gs} and C_{ds} . Besides, layout design may probably aggravate this problem. Because of auto-zeroing technique, a plate of C_i switches between node X and the amplifier input, which leads to the fact that C_p cannot be simply considered as a part of C_i as in a conventional differential

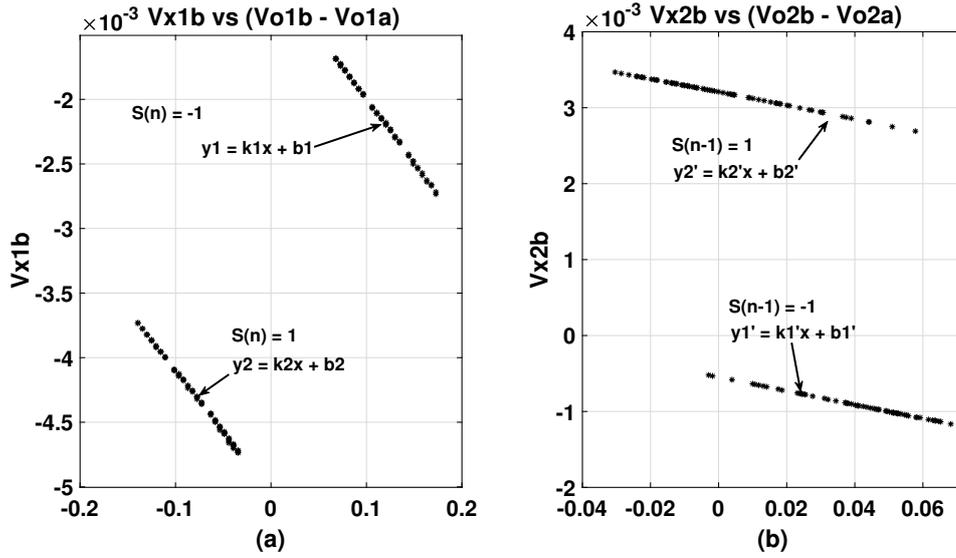


Figure 5.5: $V_{x,b}$ in the function of difference between the integrator outputs in two phases $V_{o,b} - V_{o,a}$

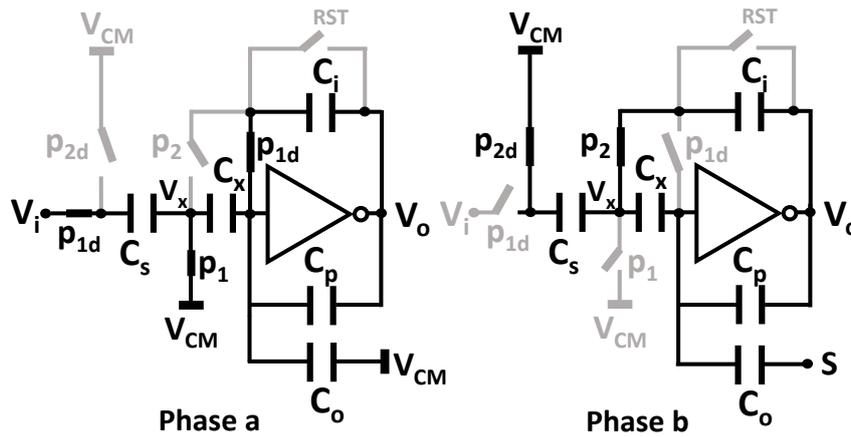


Figure 5.6: An inverter-based SC integrator with parasitic capacitors

integrator. The existence of C_p creates a path for the charge, so the charge in C_x varies between different phases. As a consequence, the potential at node X is no longer constant. The impact of variation of this potential will be elaborated later.

In addition to the fact that $V_{x,b}$ seems to be a linear function of $V_{o,b} - V_{o,a}$, in Fig. 5.5, we can also discover that $V_{x,b}$ is grouped according the modulator output S . So we chose to model that effect as another parasitic capacitor C_o . As shown in Fig. 5.6, one plate of it is connected to amplifier input while the other plate is connected to comparator output. Due to the synchronous character of the comparator, its output shifts from V_{cm} to the comparison results (-1 or 1) when the phase of the first integrator changes from a to b; while for the second integrator this happens when it changes from phase b to phase a. As there is no direct connection between the amplifier input

and the comparator output in modulator schematic, C_o is completely brought by layout design. Because of C_o , the values of integrators are modulated by their outputs, namely inter-modulation, to some extent. Also it has contribution to the variation of the potential V_x .

5.2.3 Circuit-level models of integrators

As mentioned previously, because of the existence of parasitic capacitors, the potential at node X does not remain constant. And this value is an important intermediate variable to deduce the relationship between the integrator input and output. Besides, due to the order of activities of the integrators and the comparator, the expressions of these two integrators may be slightly different. Here we define C_{s1} , C_{x1} , C_{i1} , C_{p1} , C_{o1} and C_{s2} , C_{x2} , C_{i2} , C_{p2} , C_{o2} as the names of capacitors respectively in the first integrator and the second one.

First integrator

During the cycle n , for the first integrator, both phases a and b occur in the current cycle. Since phase b of the first integrator is after the comparison, the value of the modulator output S acting on one plate of C_{o1} is the value of the current cycle $S(n)$. According to the fact that from phase a to phase b, we have $\Delta Q_{p1} + \Delta Q_{o1} = -\Delta Q_{x1}$, the potential of node X of the first integrator during phase b $V_{x1,b}(n)$ can be deduced as:

$$\begin{aligned} V_{x1,b}(n) = & -\frac{C_{p1}}{C_{x1}}(V_{o1,b}(n) - V_{o1,a}(n)) - \frac{C_{o1}}{C_{x1}}S(n)V_{ref} \\ & -\left(1 + \frac{C_{p1}}{C_{x1}} + \frac{C_{o1}}{C_{x1}}\right)\left(\frac{V_{o1,b}(n)}{A_{1,b}(n)} - \frac{V_{o1,a}(n)}{A_{1,a}(n)}\right) \\ & +\left(1 + \frac{C_{p1}}{C_{x1}} + \frac{C_{o1}}{C_{x1}}\right)\left(\frac{1}{A_{1,b}(n)} - \frac{1}{A_{1,a}(n)}\right)V_{of1} \end{aligned} \quad (5.11)$$

where $A_{1,a}(n)$, $A_{1,b}(n)$ are respectively algebraic values of gain of amplifier and V_{of1} is offset at the input of amplifier in the first integrator.

Then according to the fact that $\Delta Q_{s1} = -(\Delta Q_{x1} + \Delta Q_{i1})$, the relationship between $V_{o1,b}(n)$ and $V_{i1}(n)$, $V_{o1,a}(n)$ can be expressed as:

$$V_{o1,b}(n) = f_{a1,b}(n)V_{o1,a}(n) + f_{i1,b}(n)V_{i1}(n) + f_{s1,b}(n)S(n)V_{ref} + f_{o1,b}(n)V_{of} \quad (5.12)$$

where $V_{o1,a}(n)$, $V_{o1,b}(n)$ are respectively the values of the first integrator in phase a and b, and $V_{i1}(n)$ is its input, at cycle n . Besides, the expressions of $f_{a1,b}(n)$, $f_{i1,b}(n)$, $f_{s1,b}(n)$, $f_{o1,b}(n)$ are respectively:

$$f_{i1,b}(n) = \frac{\frac{C_{s1}}{C_{i1}}}{f_{b1,b}(n)} \quad (5.13)$$

$$f_{a1,b}(n) = \frac{C_{i1} + C_{p1}(1 + \frac{C_{i1}}{C_{x1}} + \frac{C_{s1}}{C_{x1}})(\frac{1}{A_{1,a}(n)} + 1)}{f_{b1,b}(n)} + \frac{(2C_{i1} + C_{s1})\frac{1}{A_{1,a}(n)} + C_{o1}(1 + \frac{C_{i1}}{C_{x1}} + \frac{C_{s1}}{C_{x1}})\frac{1}{A_{1,a}(n)}}{f_{b1,b}(n)} \quad (5.14)$$

$$f_{o1,b}(n) = -\frac{C_{i1}(1 + \frac{1}{A_{1,a}(n)}) + C_{p1}(1 + \frac{C_{i1}}{C_{x1}} + \frac{C_{s1}}{C_{x1}})(\frac{1}{A_{1,b}(n)} - \frac{1}{A_{1,a}(n)})}{f_{b1,b}(n)} + \frac{(C_{s1} + C_{i1})(\frac{1}{A_{1,b}(n)} - \frac{1}{A_{1,a}(n)})}{f_{b1,b}(n)} + \frac{C_{o1}(1 + \frac{C_{i1}}{C_{x1}} + \frac{C_{s1}}{C_{x1}})(\frac{1}{A_{1,b}(n)} - \frac{1}{A_{1,a}(n)})}{f_{b1,b}(n)} \quad (5.15)$$

$$f_{s1,b}(n) = \frac{\frac{C_{o1}}{C_{x1}}}{f_{b1,b}(n)} \quad (5.16)$$

where:

$$f_{b1,b}(n) = C_{i1} + C_{p1}(1 + \frac{C_{i1}}{C_{x1}} + \frac{C_{s1}}{C_{x1}})(\frac{1}{1 - A_{1,b}(n)}) + (C_{i1} + C_{s1})\frac{1}{A_{1,b}(n)} + C_{o1}(1 + \frac{C_{i1}}{C_{x1}} + \frac{C_{s1}}{C_{x1}})\frac{1}{A_{1,b}(n)} \quad (5.17)$$

These coefficients are modulated by the non-linear amplifier gain, which depends on the integrator output. As a consequence, they are time-varying.

We already know that $V_{o1,b}(n)$ directly depends on $V_{o1,a}(n)$, now we want to understand the relationship between $V_{o1,a}(n)$ and $V_{o1,b}(n - 1)$, in order to understand the complete function of integrator. From phase b of the previous cycle ($n - 1$) to phase a of cycle n , similarly, we have $\Delta Q_{x1} = -(\Delta Q_{p1} + \Delta Q_{o1} + \Delta Q_{i1})$, thus the expression of $V_{o1,a}(n)$ is:

$$V_{o1,a}(n) = f_{a1,a}(n)V_{o1,a}(n - 1) + f_{b1,a}(n)V_{o1,b}(n - 1) + f_{s1,a}(n)S(n - 1)V_{ref} + f_{o1,a}(n)V_{of1} \quad (5.18)$$

with:

$$f_{a1,a}(n) = -\frac{\frac{C_{p1}}{C_{x1}}(C_{i1} + C_{x1})(1 + \frac{1}{A_{1,a}(n)}) - (1 + \frac{C_{o1}}{C_{x1}})(C_{i1} + C_{x1})\frac{1}{A_{1,a}(n)}}{f_{a01,a}(n)} \quad (5.19)$$

$$f_{b1,a}(n) = \frac{C_{i1}(\frac{C_{p1}}{C_{x1}} + 1) + C_{i1}(\frac{C_{p1}}{C_{x1}} + \frac{C_{o1}}{C_{x1}})\frac{1}{A_{1,b}(n)} + (C_{i1} + 2C_{x1})\frac{1}{A_{1,b}(n)}}{f_{a01,a}(n)} \quad (5.20)$$

$$f_{o1,a}(n) = \frac{C_{i1}(1 + \frac{1}{A_{1,a}(n)}) + (C_{o1} + C_{p1} + C_{x1})(\frac{1}{A_{1,a}(n)} - \frac{1}{A_{1,b}(n)})}{f_{a01,a}(n)} - \frac{(C_{x1} + C_{i1})(1 + \frac{C_{p1}}{C_{x1}} + \frac{C_{o1}}{C_{x1}})(\frac{1}{A_{1,b}(n-1)} - \frac{1}{A_{1,a}(n-1)})}{f_{a01,a}(n)} \quad (5.21)$$

$$f_{s1,a}(n) = \frac{\frac{C_{o1}}{C_{x1}}(C_{i1} + C_{x1})}{f_{a01,a}(n)} \quad (5.22)$$

where:

$$f_{a01,a}(n) = C_{i1} + C_{p1}(1 + \frac{1}{A_{1,a}(n)}) + (C_{i1} + C_{o1} + C_{x1})\frac{1}{A_{1,a}(n)} \quad (5.23)$$

Combining equation (5.12) and equation (5.18) and we have:

$$\begin{aligned} V_{o1,b}(n) &= f_{i1,b}(n)V_{i1}(n) + f_{a1,b}(n)f_{b1,a}(n)V_{o1,b}(n-1) \\ &\quad + f_{s1,b}(n)S(n)V_{ref} + f_{a1,b}(n)f_{s1,a}(n)S(n-1)V_{ref} \\ &\quad + f_{o1,b}(n)V_{of1} + f_{a1,b}(n)f_{o1,a}(n)V_{of1} \\ &\quad + f_{a1,b}(n)f_{a1,a}(n)V_{o1,a}(n-1) \end{aligned} \quad (5.24)$$

Thus we can see that due to $f_{a1,a}(i) \neq 0, (i = 1, \dots, n)$, the value of integrator $V_{o1,b}(n)$ does not only depend on the input and its last state but also all its previous states. In addition to this, $V_{o1,b}(n)$ is modulated by the modulator outputs. Also there is still a term of offset which indicates an effectiveness loss of the auto-zeroing technique.

Second integrator

The thought to deduce the expression of the value of the second integrator is the same. The value of the second integrator at its phase b can also be written in the form of equation (5.24) with time-varying coefficients $f_{a2,b}(n), f_{i2,b}(n), f_{s2,b}(n), f_{o2,b}(n)$ and $f_{a2,a}(n), f_{b2,a}(n), f_{s2,a}(n), f_{o2,a}(n)$.

However, because of the delay of one phase (as shown in Fig. 3.6), a period including phase a and phase b of the second integrator crosses cycle $(n - 1)$ and cycle n of modulator. (The time of $V_{o2,a}(n)$ is the same moment of $V_{o1,b}(n - 1)$ and the time of $V_{o2,b}(n)$ is the same of $V_{o1,a}(n)$.) As the second integrator integrates before the comparison, the value of the modulator output S acting on one side of C_{o2} is the value of last cycle, namely $S(n - 1)$. The potential at node X in phase b of the second integrator in cycle n , $V_{x2,b}$, can be expressed as:

$$\begin{aligned}
V_{x2,b}(n) = & -\frac{C_{p2}}{C_{x2}}(V_{o2,b}(n) - V_{o2,a}(n)) + \frac{C_{o2}}{C_{x2}}S(n - 1)V_{ref} \\
& -\left(1 + \frac{C_{p2}}{C_{x2}} + \frac{C_{o2}}{C_{x2}}\right)\left(\frac{V_{o2,b}(n)}{A_{2,b}(n)} - \frac{V_{o2,a}(n)}{A_{2,a}(n)}\right) \\
& +\left(1 + \frac{C_{p2}}{C_{x2}} + \frac{C_{o2}}{C_{x2}}\right)\left(\frac{1}{A_{2,b}(n)} - \frac{1}{A_{2,a}(n)}\right)V_{of2}
\end{aligned} \tag{5.25}$$

The form of $V_{x2,b}(n)$ is almost the same as that of $V_{x1,b}(n)$, except the sign of the term $S(n - 1)\frac{C_{o2}}{C_{x2}}$. As a consequence, the signs of $f_{s2,a}(n)$ and $f_{s2,b}(n)$ are changed:

$$f_{s2,a}(n) = -\frac{\frac{C_{o2}}{C_{x2}}(C_{i2} + C_{x2})}{f_{a02,a}(n)} \tag{5.26}$$

$$f_{s2,b}(n) = -\frac{\frac{C_{o2}}{C_{x2}}}{f_{b2,b}(n)} \tag{5.27}$$

Except this, expressions of other coefficients refer to those of the first integrator. Similarly, to the first integrator, we have:

$$\begin{aligned}
V_{o2,b}(n) = & f_{i2,b}(n)V_{i2}(n) + f_{a2,b}(n)f_{b2,a}(n)V_{o2,b}(n - 1) \\
& +f_{s2,b}(n)S(n - 1)V_{ref} + f_{a2,b}(n)f_{s2,a}(n)S(n - 1)V_{ref} \\
& +f_{o2,b}(n)V_{of2} + f_{a2,b}(n)f_{o2,a}(n)V_{of2} \\
& +f_{a2,b}(n)f_{a2,a}(n)V_{o2,a}(n - 1)
\end{aligned} \tag{5.28}$$

If we apply $C_{pi} = C_{oi} = 0$, $(i = 1, 2)$ and infinite amplifier gain, the expressions of $V_{o1,b}(n)$ and $V_{o2,b}(n)$ return to the equation (3.22) which is for an ideal inverter-based SC integrator. Besides, equation (5.11) and equation (5.25) well explicate the phenomenon of $V_{o,b}$ observed in Fig. 5.4.

5.3 Validation of the new proposed inverter-based SC integrator models

In last section, we proposed new models of the integrators in the second order $\Sigma\Delta$ Modulator, so now we want to verify our new models with the help of post-layout simulation of the $\Sigma\Delta$ Modulator. We record the values of the signals in equation (5.24) and (5.28). According to these values, we could estimate the parameters of integrators. Then we re-calculated the values of two integrator outputs at each clock cycles using equations (5.24) and (5.28). On the other hand, with the record signal values, the theoretical outputs of an ideal integrator at each clock cycle could also be calculated using equation (3.22). We also looked at the errors between the re-calculated integrator values and those in post-layout simulation as well as the errors of the theoretical models. If the errors of the new integrator models decrease, we can say that the new proposed integrator models match better than the theoretical ones.

5.3.1 Parameters identification for the new proposed inverter-based SC integrator models

We launched a post-layout simulation in one step with $M_1 = 8$ and record the values of signals in equations (5.24) and (5.28). According to the values of $V_{x1,b}$ and $V_{x2,b}$ in post-layout simulation, the values of C_{pm}/C_{xm} and C_{om}/C_{xm} , ($m = 1, 2$) can be estimated based on equations (5.11) and (5.25) by substituting an infinite amplifier gain. By researching least-mean-square of the difference between the values of $V_{x,b}$ saved in post-layout simulation and the values estimated using equations (5.11) and (5.25), we obtain $C_{p1}/C_{x1} = 0.0099$, $C_{p2}/C_{x2} = 0.0086$, $C_{o1}/C_{x1} = 0.0023$, $C_{o2}/C_{x2} = 0.0022$. The values of C_o are relatively small compared to those of C_p . As illustrated, C_o is completely caused by layout design while C_p exists systematically in an inverter-based SC integrator and it may be aggravated by layout design.

5.3.2 Comparison of the new proposed inverter-based SC integrator models to the ideal integrator model

From equations (5.13) to (5.17), we find that in the new proposed integrator models, non-linear amplifier gain appears in these equations in the form of reciprocal of gain or difference of reciprocal of gain, since the values of the amplifier gain are relatively great, their impacts can be ignored. For this reason, we supposed that the amplifier gain was infinite once again. In addition,

we supposed that there is no mismatch between C_s , C_i and C_x . Thus the values of the coefficients in equations (5.24) and (5.28) could be calculated through equations (5.13) to (5.17). Then the values of integrator outputs at each clock cycle could be estimated through equations (5.24) and (5.28). We defined the estimation errors as the difference between the estimated integrator outputs using certain integrator model and the integrator output values record in post-layout simulation. Thus the estimation errors of the new integrator models based on equations (5.24) and (5.28) and the estimation errors of the theoretical integrator model based on equation (3.22) could be respectively calculated.

Fig. 5.7a shows the errors between the ideal integrator model and the model in post-layout simulation in the first integrator while Fig. 5.7b shows those between the new proposed integrator model and that in post-layout simulation. We find that comparing to the ideal model, the proposed model matches better, since its errors are more concentrated. If we optimal the parameters by researching least-mean-square of the errors, the magnitude of the estimated errors in the new model decreases to 10^{-4} , as shown Fig. 5.7c. In post-layout simulation, in addition to the parasitic capacitors, there is a mismatch between C_{s1} , C_{i1} and C_{x1} , and we cannot estimate it based on Fig. 5.5. That leads to the fact that the parameters cannot be estimated precisely. That why absolute values of the errors in Fig. 5.7b are still significant. However, capacitor mismatch has been well considered for the optimal parameters. In fact, the values of parameters estimated and optimal parameters are very close. Surprisingly, even the little change in parameter values causes large variation in errors.

The same observation can be obtained according Fig. 5.8a and Fig. 5.8b for the second integrator. To be pay attention that in Fig. 5.8a, the estimation errors are separated into two groups. It is coherent to equation (5.28) that $V_{o2,b}(n)$ is only correlated to $S(n-1)$. It indicates that the new proposed model of inverter-based SC integrators is correct in our circuit. Of course, we are not able to further decrease the estimation errors as we don't know exactly the values of the amplifier gain at each clock cycle.

5.4 Validation of the $\Sigma\Delta$ modulator model composed of the new proposed inverter-based SC integrator models

The objective of proposing a new integrator model is to build a modulator model finally in order to explicate, as much as possible, its behaviors and degradation in post-layout simulation.

After validating the new models at integrator level, we then completed the modulator model

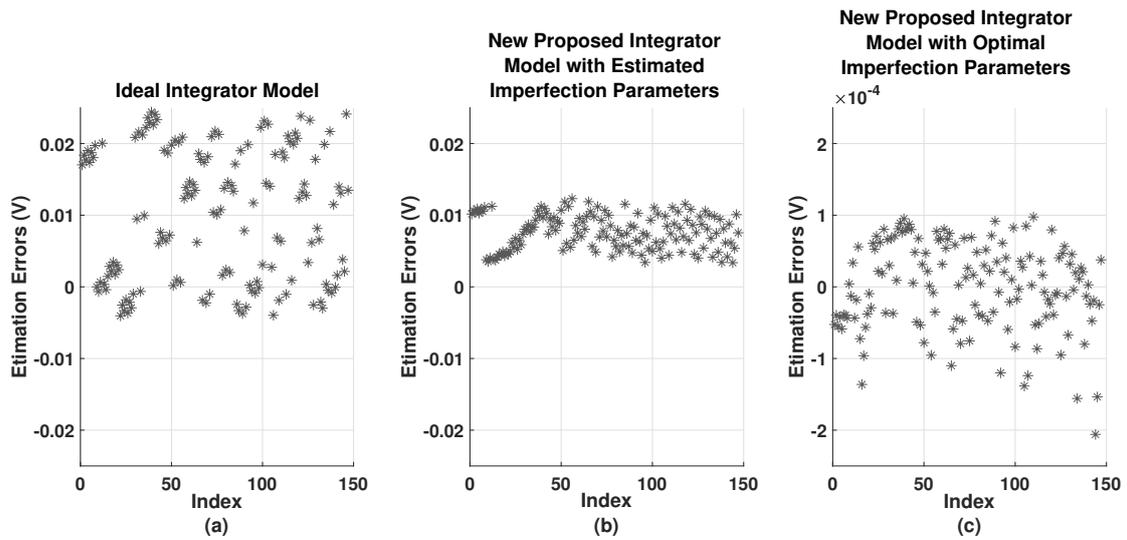


Figure 5.7: Comparison of the estimation errors in the first integrator respectively of the ideal integrator model and of the new proposed model.

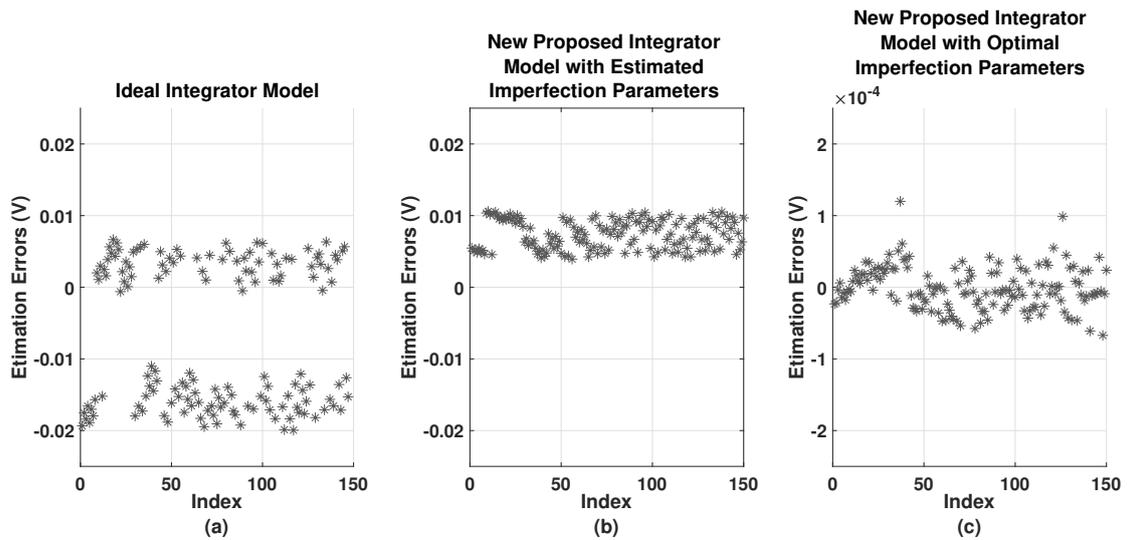


Figure 5.8: Comparison of the estimation errors in the second integrator respectively of the ideal integrator model and of the new proposed model.

by adding other circuit imperfections. Then we identified the parameters of other imperfections through post-layout simulation too. Apart from the signals in equations (5.24) and (5.28), we also record the values of certain signals in the $\Sigma\Delta$ modulator (including S/H) and estimated the values of other defect parameters.

For the validation of the new $\Sigma\Delta$ modulator model, we compared not only ER as well as the forms of conversion errors in several scenes. If both ER and the forms of conversion errors are similar, we can consider that the new modulator model is reasonable.

5.4.1 Parameters identification of other imperfections in $\Sigma\Delta$ modulator

For other imperfection parameters, it can be directly estimated by observing the signal waveform in post-layout simulation: $V_{of1} \approx V_{of2} = -0.021V$ and offset of the threshold voltage of comparator $V_{th} = 0.027V$. Apart of this, we model a linear error in the S/H block: $X = k_{sh}V_{pixel} + b_{sh}$, where V_{pixel} is the input of S/H block and X is its output as well as the input of modulator. By recording the inputs and outputs of the S/H block, k_1 and b are estimated with the method of linear regression.

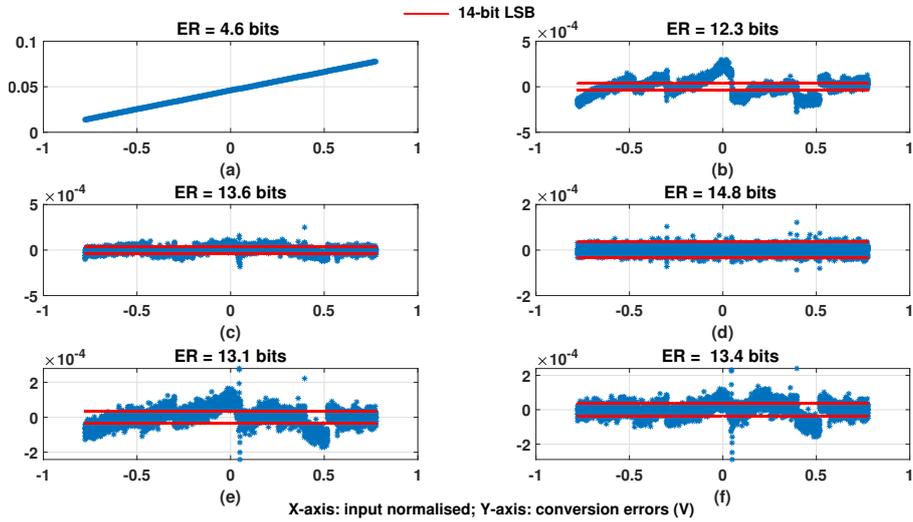
5.4.2 Comparison of proposed modulator model and modulator in post-layout simulation

In one-step mode

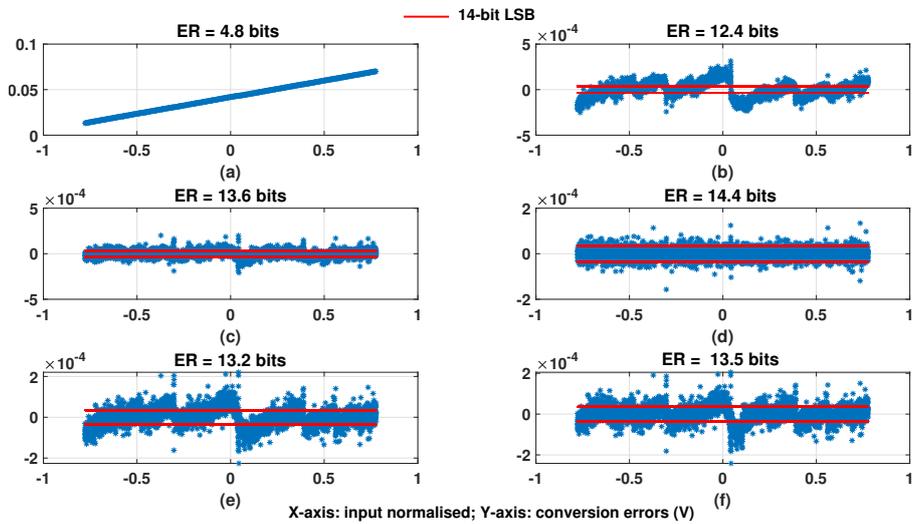
The conversion mode is firstly chosen for comparison is the one-step conversion, as there is no inter-stage errors. Fig. 5.9A respectively shows the forms of conversion error in different scenes in post-layout simulation: without correction (with theoretical filter), corrected gain and offset, corrected with optimal filter and corrected with pattern-correcting filter with $n = 3$, simplified filter in plan A and hybrid filter with $n = 3, m = 3$ in plan A. To be noticed that it is a fourth-order IIR applied in simplified filter and the hybrid filter. Even in last Chapter, we found that a second-order IIR was enough for two-step mode conversions, it didn't work well for a long conversion in one-step mode. In stead, a higher-order IIRs seems to work. However, ER won't be further improved if we continue increasing IIR filter order. The correction results in Fig. 5.9A will be considered as the reference to judge whether the models are reasonable. Then some simulations of the model with estimated parameters of this 2^{nd} $\Sigma\Delta$ modulator have also been done with the same condition: $M = 400$.

Simulation with infinite amplifier gain

In the first simulation, we chose the amplifier gain $A = 120\text{dB}$ for both integrators in order to simulate the ideal amplifiers. Therefore, there is only linear error in conversion errors. Fig. 5.9B shows the form of conversion errors in the scenes mentioned previously in the simulation of the modulator model. For the first three cases, almost the same ERs are obtained compared to that of post-layout simulation. In addition to it, the forms of errors are similar to those of post-layout simulations in Fig. 5.9A. It proves that the proposed high-level modulator model is close to the circuit in post-layout. From the comparison, we can consider that in the circuit in post-layout



(A) In post-layout simulation: (a) with theoretical filter; (b) with correction of gain and offset; (c) with optimal filter correction; (d) with pattern-correcting filter correction of 3-bit package ($n = 3$); (e) with simplified filter; (f) with hybrid filter of 3-bit package, $m = 3$



(B) In simulations of proposed modulator model with infinite amplifier gain: (a) with theoretical filter; (b) with correction of gain and offset; (c) with optimal filter correction; (d) with pattern-correcting filter correction of 3-bit package ($n = 3$); (e) with simplified filter; (f) with hybrid filter of 3-bit package, $m = 3$

Figure 5.9: Conversion errors in one-step mode.

simulation, most part of error corrected with these coarse corrections is the error caused by C_{px} and C_{ox} , ($x = 1, 2$). In other words, the real non-linear error caused by amplifier gain is still hiding behind the linear error. That can explicate why the improvement brought by the pattern-correcting filter is less in the proposed model compared to post-layout. Besides, we should pay attention to the fact that the simplified filter correction and hybrid filter correction work a little bit better in the proposed model, probably because we didn't model the non-linear amplifier gain. As a result, it is able to approximate the impulse response of the optimal filter more precisely. Apart from non-linear amplifier gain, we didn't model the error caused by the different signal path as illustrated in Chapter 3. That why the hybrid filter improves scarcely ER on the basis of the simplified filter in the proposed model. We should also pay attention that the simplified filter correction and the hybrid filter correction cannot improve ERs to 14 bits neither for the model in post-layout simulation nor for the proposed model as 400 coefficients are too many to approximated with a IIR.

Simulation with non-linear finite amplifier gain

In this simulation, we add the non-linear finite amplifier gain as introduced in Section 3 to the proposed model. Here we choose $A_{max} = 75$ dB. Fig. 5.10 shows respectively conversion errors with four different ways to reconstruct the modulator input. We hardly see the change in the forms of errors for the first four cases. And exceptionally, there is about 0.1-bit increase of ER in these cases. Since the change isn't significant we can consider that with this non-linear amplifier gain model (the approximation of polynomial), the non-linear amplifier gain doesn't make significant difference to the ADC. However, with a finite and non-linear gain, the results of the simplified filter correction and hybrid filter correction are degraded because the non-linear gain increases difficulty to approximate the impulse response.

In two-step mode

We also have comparison in two-step mode with $M_1 = 48, M_2 = 24$ keeping the same parameters of model and choosing $A_{max} = 120$ dB. Table 4.1 shows the correction results of this model comparing to those in post-layout simulation. Even though there are decreases in ER for all cases in two-step mode, the variation is about 0.1 bits and it can be ignored. Except it, the trend is similar to that in one-step conversion: with the existing correction methods, results are similar for both models. While with the pattern-correcting filter, the improvement in ER on the basis of the optimal filter is less in the proposed model compared to post-layout. Furthermore, the simplified filter correction and hybrid filter correction work better for the proposed model as there are less

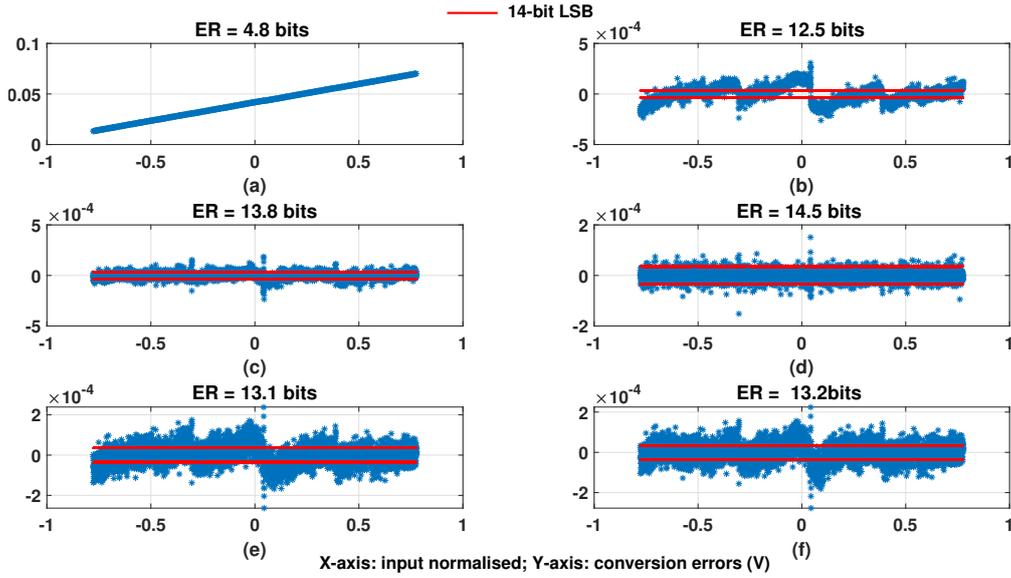


Figure 5.10: Conversion error in one-step mode in functional model simulations with non-linear finite amplifier gain: (a) with theoretical filter; (b) with correction of gain and offset; (c) with optimal filter correction; (d) with pattern-correcting filter correction of 3-bit package ($n = 3$); (e) with simplified filter; (f) with hybrid filter of 3-bit package, $m = 3$

Table 5.1: Comparison of correction results of the proposed model in two-step mode to post-layout simulation

Correction methods	ER (bits)		
	Post-layout	proposed model, $A_{max} = 120dB$	proposed model, $A_{max} = 75dB$
Theoretical filter	4.9	4.8	4.8
Correction gain and offset	12.2	12.2	12.3
Optimal filter, plan A	14.2	14.1	14.0
Optimal filter, plan B	14.3	14.5	14.3
Pattern-correcting filter, $m = 3$, plan A	15.0	14.2	14.1
Pattern-correcting filter, $m = 3$, plan B	15.2	14.7	14.6
Simplified filter, plan A	13.7	14.1	14.0
Hybrid filter, $m_{bit} = 3, m = 3$ plan A,	13.9	14.1	14.0

errors sources in the proposed model than the model in post-layout simulation.

Fig. 5.11 shows form of conversion errors in two-step mode in post-layout simulation while those in the simulation of functional model respectively with infinite amplifier gain and non-linear finite amplifier gain are shown in Fig. 5.12 and Fig. 5.13. To be mentioned that in terms of the shapes of conversion errors in two-step model, they are not enough similar compared to one-step mode. Maybe it is because that we didn't model the inter-step error which may occur during the sampling of residue at the end of the first step.

According to the comparison between the correction results of circuit in post-layout and the proposed model, we can obtain some acknowledges about our circuit as well as correction meth-

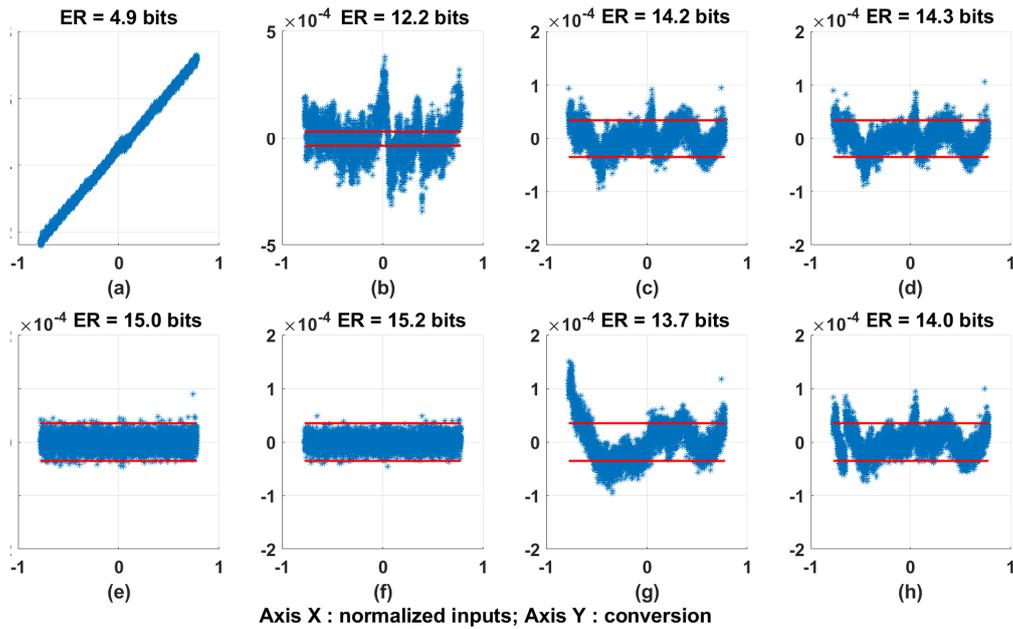


Figure 5.11: Conversion errors in two-step mode in post-layout simulation: (a) with theoretical filter; (b) with correction of gain and offset; (c) with optimal filter correction in plan A; (d) with optimal filter correction in plan A; (e) with pattern-correcting filter correction of 3-bit package ($n = 3$) in plan A; (f) with pattern-correcting filter correction of 3-bit package ($n = 3$) in plan B; (g) with simplified filter in plan A; (h) with hybrid filter with $n = 3, m = 3$ in plan A.

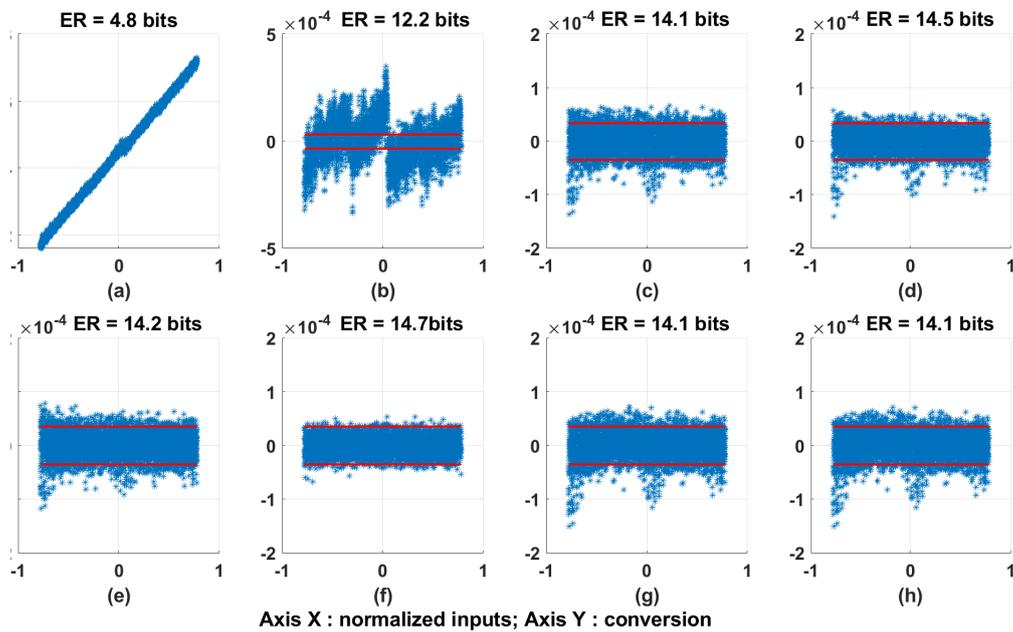


Figure 5.12: Conversion errors in two-step mode in functional model simulations with infinite amplifier gain: (a) with theoretical filter; (b) with correction of gain and offset; (c) with optimal filter correction in plan A; (d) with optimal filter correction in plan A; (e) with pattern-correcting filter correction of 3-bit package ($n = 3$) in plan A; (f) with pattern-correcting filter correction of 3-bit package ($n = 3$) in plan B; (g) with simplified filter in plan A; (h) with hybrid filter with $n = 3, m = 3$ in plan A.

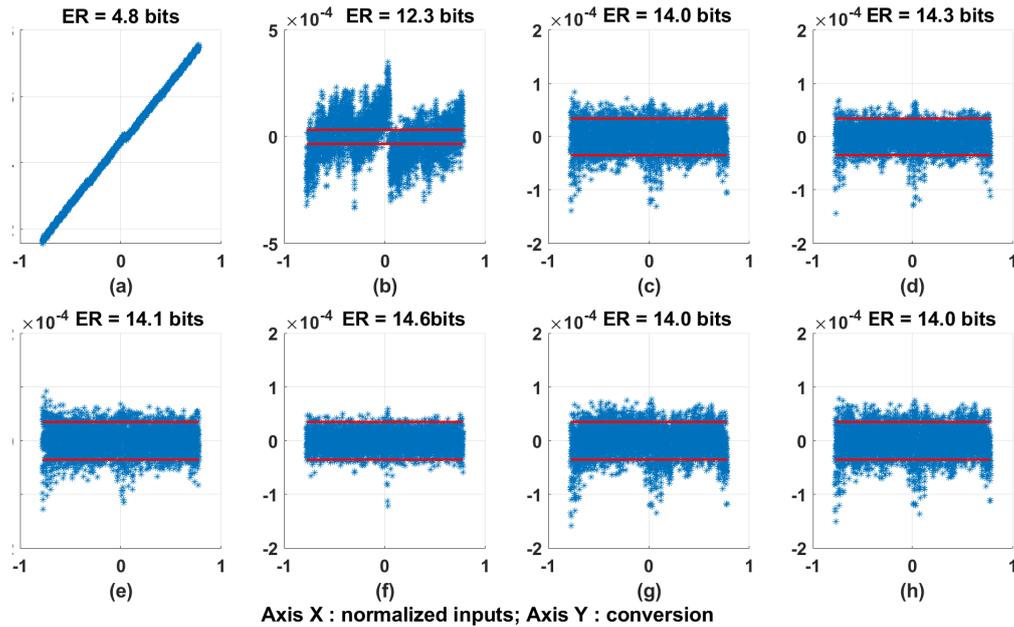


Figure 5.13: Conversion errors in two-step mode in functional model simulations with non-linear finite amplifier gain: (a) with theoretical filter; (b) with correction of gain and offset; (c) with optimal filter correction in plan A; (d) with optimal filter correction in plan A; (e) with pattern-correcting filter correction of 3-bit package ($n = 3$) in plan A; (f) with pattern-correcting filter correction of 3-bit package ($n = 3$) in plan B; (g) with simplified filter in plan A; (h) with hybrid filter with $n = 3, m = 3$ in plan A.

ods:

1. The most of degradation can be explicated by parasitic capacitors and these errors can be well corrected with the optimal filter.

2. Compared with the optimal, the advantage of the pattern-correcting filter is that it is able to correct the fine errors (whose amplitude is equivalent to LSB less than 13 bits) in the circuit. It is shown in post-layout simulation that it improves ER further than the optimal filter.

3. For the fact that the improvements brought about by the pattern-correcting filter are less in the proposed model, we consider that there are other error sources we didn't discovered in the post-layout level. For example, the errors correlated to the current and its two adjacent modulator output bits in S/H shown in Fig. 4.11

4. For few clock cycles, the correction effects of the simplified filter or the hybrid filter is comparable to that of the optimal filter while they require much less coefficients. However, for a long conversion, the simplified filter and the hybrid filter don't work as well as they do in a short conversion because the FIR impulse response cannot be approximated by an IIR precisely, especially the MSBs in the impulse response of FIR. It is necessary to increase the order of IIRs, in one-step mode with 400 clock cycles.

5. Although we cannot find out all the defects in the circuit in post-layout, the proposed high-

level integrator model can explicate most part of the ADC degradation in post-layout simulations. With this model, we can research deep into the correction methods.

5.4.3 Study of the parasitic capacitor impacts on correction methods

In the last sub-section, we just validated the proposed model and obtained a conclusion: the ADC degradation in post-layout simulations is mainly caused by C_p and C_o . Now we want to find the maximum values of parasitic capacitor of the proposed model, with which ER can stay over 14 bits with the certain correction methods. To be noticed that for the modulator model, we suppose that there isn't any other defects except parasitic capacitors.

In one-step mode

Fig. 5.14(a) shows ERs respectively after correction of the optimal filter and the pattern-correcting filter with 3-bit package in the function of C_{p1} and C_{p2} (with infinite amplifier gain). For both corrections, ER drops from nearly 18 bits to about 11 bits with increase of C_{p1} and C_{p2} . That means that ADC resolution is very sensitive to C_p . To guarantee 14 bits, $C_{p1}/C_{x1} + C_{p2}/C_{x2}$ should less than 0.3. However, the pattern-correcting filter doesn't show significant advantage in ER compared to the optimal filter for the proposed model. It is because we didn't model all error sources in the circuit and the optimal filter is capable of correcting most of errors caused by C_p .

Fig. 5.14(b) shows ER in the function of C_{o1} and C_{o2} with $C_{p1}/C_{x1} = C_{p2}/C_{x2} = 0.01$ (this value is close to that in our circuit in the post-layout simulation), with the same corrections. We observe that ERs varies around 17.6 bits with the pattern-correcting filter in spite of the variation of C_{o1} and C_{o2} while with the optimal filter, ERs are about 17.3 bits. It indicates that although the impact of C_o is irregular, it can be ignored after correction.

Fig. 5.15 (a) and (b) show ERs using theoretical filter and with the optimal filter corrections and the simplified filter correction. The reason why we add the theoretical filter in this simulation is the theoretical filter can also be expressed with a IIR filter in z domain. As shown in Fig. 5.15 (a) we find that these correction filters are sensitive to C_p too. Despite of the value of C_p , the simplified filter always provides better ER than the theoretical filter. And compared to the optimal filter, the simplified filter provides almost the same ER with much more less coefficients. However, according to Fig. 5.15 (b), C_o has little impact on the correction results. But we should realize that, with theoretical filter, ER is extremely sensitive to C_{o2} .

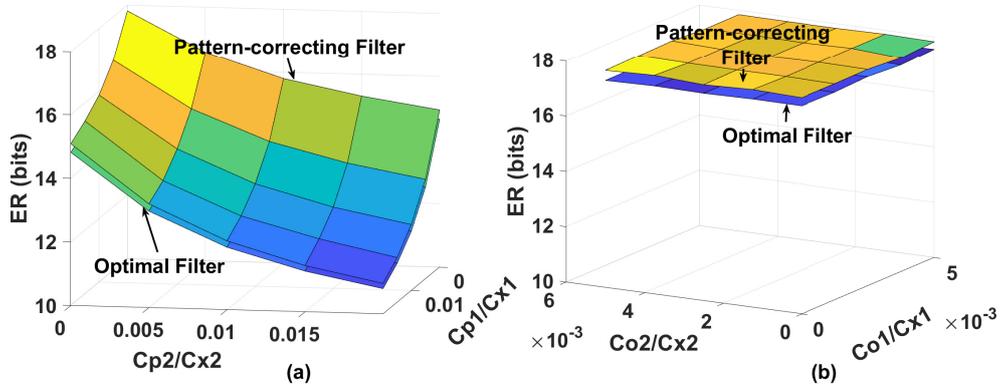


Figure 5.14: Equivalent resolution in one-step mode respectively with the optimal filter correction and the pater-correcting filter correction (package of 3 bits): (a) sweep of C_{p1} and C_{p2} and $C_{o1} = C_{o2} = 0$; (b)sweep of C_{o1} and C_{o2} and $C_{p1}/C_{x1} = C_{p2}/C_{x2} = 0.01$;

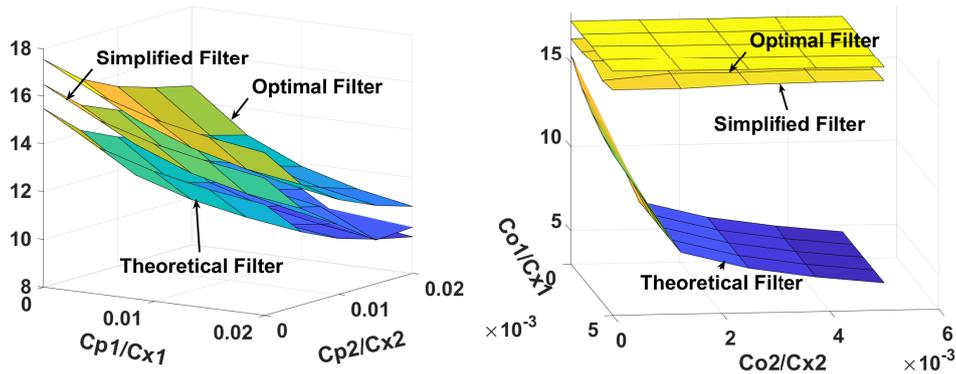


Figure 5.15: Equivalent resolution in one-step mode respectively with theoretical filter and with the optimal filter correction and the simplified filter correction (package of 3 bits): (a) sweep of C_{p1} and C_{p2} and $C_{o1} = C_{o2} = 0$; (b)sweep of C_{o1} and C_{o2} and $C_{p1}/C_{x1} = C_{p2}/C_{x2} = 0.01$;

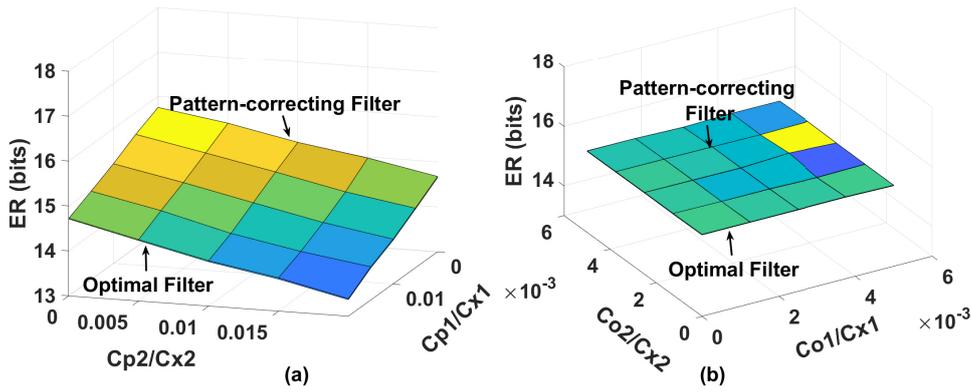


Figure 5.16: Equivalent resolution in two-step mode respectively with the optimal filter correction and the patter-correcting filter correction (package of 3 bits): (a) sweep of C_{p1} and C_{p2} and $C_{o1} = C_{o2} = 0$; (b)sweep of C_{o1} and C_{o2} and $C_{p1}/C_{x1} = C_{p2}/C_{x2} = 0.01$;

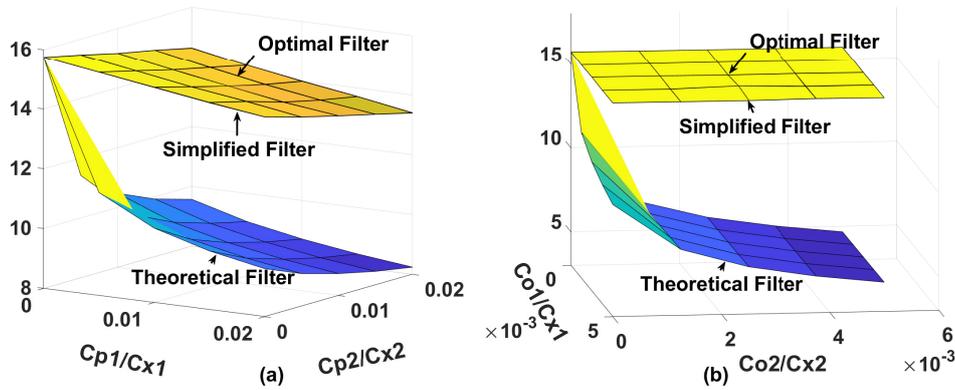


Figure 5.17: Equivalent resolution in two-step mode respectively with theoretical filter and with the optimal filter correction and the simplified filter correction (package of 3 bits): (a) sweep of C_{p1} and C_{p2} and $C_{o1} = C_{o2} = 0$; (b)sweep of C_{o1} and C_{o2} and $C_{p1}/C_{x1} = C_{p2}/C_{x2} = 0.01$;

In two-step mode

As shown in Fig. 5.16(a) and (b) show ERs in two-step mode respectively after the correction of optimal filter and the pattern-correcting filter ($m = 3$, plan A) in the function of C_{p1} and C_{p2} . The trends are similar to those in one-step mode and ER is more stable in two-step mode as it drops only from 15.8 bits to 13.2 bits with increase of C_p after corrections. Besides, the pattern correcting filter can scarcely improve ER compared to the optimal filter. Fig. 5.17(a) and (b) show ERs in two-step mode with theoretical filter and with the optimal filter corrections and the simplified filter correction in plan A. Different to one-step mode, in two-step mode, the simplified filter is able to provide almost the same ER as the optimal filter regardless of C_p and C_o . However, without correction, ER seems to be more sensitive in two-step mode. Therefore, the simplified filter has great advantage in two-step mode.

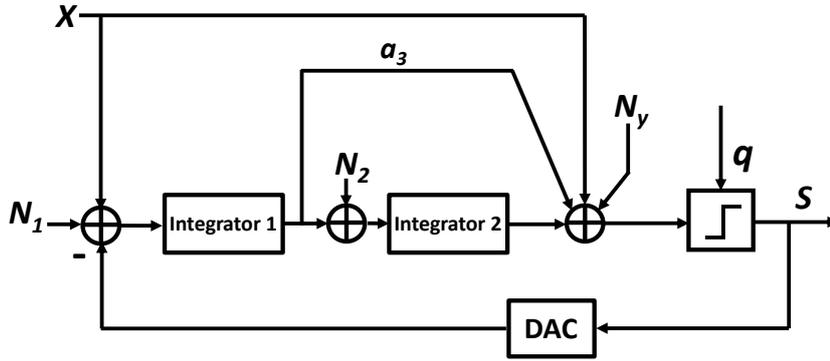


Figure 5.18: Diagram of the $\Sigma\Delta$ modulator with noise sources.

5.4.4 Study of the noise resistance of correction methods

The proposed model can help us to study the robustness of correction methods regarding noise. Here we chose the modulator model with the defects including those defined in 5.4.1 to simulate our circuit. Fig. 5.18 shows the equivalent model of the $\Sigma\Delta$ modulator with diverse noise sources. We define respectively noise sources N_1 , N_2 and N_y as the noise added at the input of the first integrator, the noise added at the input of the second integrator and that added at the input of the comparator. We consider that noise is uniformly distributed within a certain range $\pm A_{noise}$. Fig. 5.19 (a) to (d) show the ER in one-step mode after corrections of the optimal filter, the pattern-correcting filter ($m = 3$) and the simplified filter, in the simulations of the proposed model with different noise source. The parameters of the proposed model are those in Section 5.4.2. N_1 is critical because it is mixed with the modulator input and it won't be filtered by the modulator. In reality, noise may appear anywhere, so we take the results in Fig. 5.19 (d) as the reference. If A_{noise} is guaranteed below a level which is equivalent to a LSB of 12 bits (calculated by $\frac{\log_{10}(\frac{V_{in,max}^2}{3A_{noise}^2})}{0.602}$), degradation due to noise of all correction methods is slight.

Fig. 5.20 (a) to (d) show the results in two-step mode. Compared to one-step mode, conversion over two-step is more sensitive to noise. To guarantee the performance of correction methods, A_{noise} should be below to 13 bits.

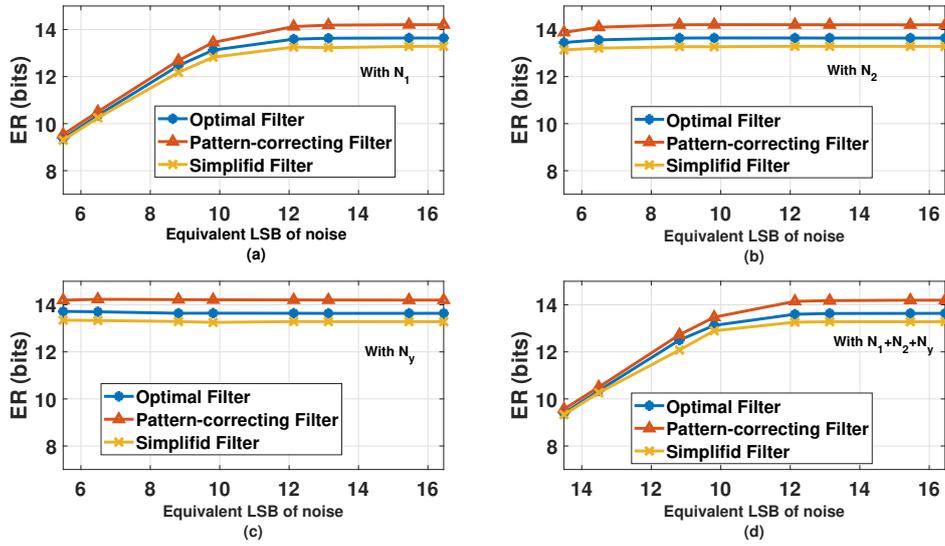


Figure 5.19: Equivalent resolution in the function of noise amplitude in one-step mode: (a) only with noise source N_1 ; (b) only with noise source N_2 ; (c) only with noise source N_y ; (d) with noise sources $N_1 + N_2 + N_y$

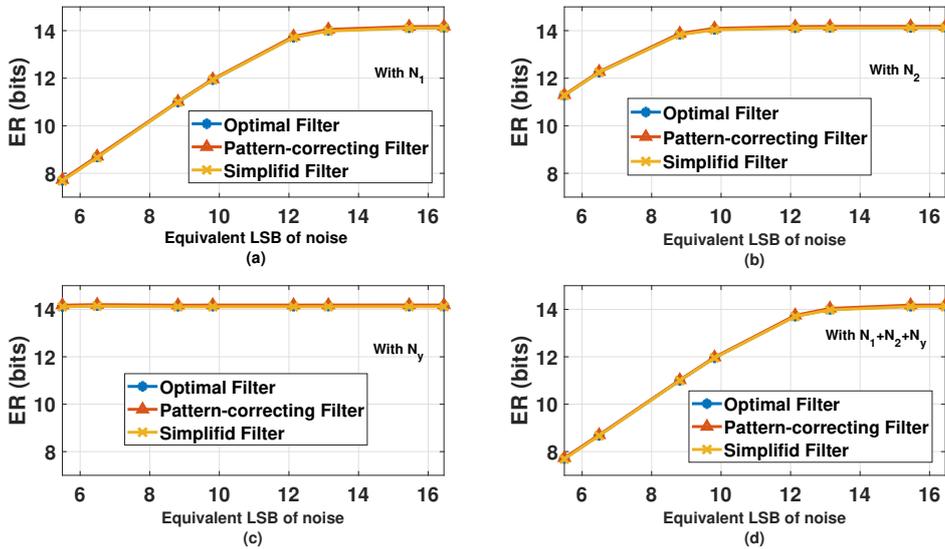


Figure 5.20: Equivalent resolution in the function of noise amplitude in two-step mode: (a) only with noise source N_1 ; (b) only with noise source N_2 ; (c) only with noise source N_y ; (d) with noise sources $N_1 + N_2 + N_y$

5.5 Further study on impact of parasitic capacitor to an inverter-based I $\Sigma\Delta$ modulator

We have already studied the effect of parasitic capacitors on the inverter-based SC integrators and on the correction of our I $\Sigma\Delta$ ADC. Since it changes the transfer function of the inverter-based SC integrators, the transfer function of the modulator must be different from the theoretical one. In this section, we will try to deduce the transfer function of the modulator after all cycles by respectively iterating equation (5.24) and equation (5.28). It helps us more intuitively understand the reason for the degradation of performance in our I $\Sigma\Delta$ ADC.

5.5.1 Deduction of transfer function of the 2nd I $\Sigma\Delta$ modulator

According to equation (5.18), $f_{a1,a} \neq 0$ is the reason why $V_{o1,a}$ cannot be completely canceled in the incremental expression of $V_{o1,b}$ as shown in equation (5.24). $f_{a1,a}$ brings a memory effect to the first integrator. In other words, the current integrator output $V_{o1,b}$ not only depends on its last value. This conclusion is also true for the second integrator.

To understand this supplementary memory effect to both integrators, we first iterate equation (5.24) and (5.18) to remove all the intermediate term $V_{o1,a}$. In this way, we obtain the relation between the current output $V_{o1,b}(n)$, V_{i1} and all its previous states $V_{o1,b}(i)$, ($i = 0, 1, \dots, n - 1$):

$$\begin{aligned}
 V_{o1,b}(n) = & f_{b1,i}(n)V_{i1}(n) + f_{b1,a}(n)f_{a1,b}(n)V_{o1,b}(n-1) \\
 & + f_{b1,a}(n) \sum_{i=0}^{n-2} f_{a1,b}(i+1)V_{o1,b}(i) \prod_{j=i+1}^{n-1} f_{a1,a}(j+1) \\
 & + f_{b1,a}(n) \sum_{i=0}^{n-1} f_{a1,s}(i+1)S(i) \prod_{j=i+1}^{n-1} f_{a1,a}(j+1) + f_{b1,s}(n)V_{ref} \\
 & + f_{b1,a}(n)V_{of1} \sum_{i=0}^{n-2} f_{a1,o}(i+1) \prod_{j=i+1}^{n-1} f_{a1,a}(j+1) + f_{b1,a}(n)f_{a1,o}(n)V_{of1} + f_{b1,o}(n)V_{of1} \\
 & + f_{b1,a}(n) \prod_{i=0}^{n-1} f_{a1,a}(i+1)V_{o1,a}(0)
 \end{aligned} \tag{5.29}$$

where $V_{o1,a}(0), V_{o1,b}(0)$ are initial states of the first integrator. Comparing equation (5.29) to equation (3.22), we find that due to $f_{a1,a} \neq 0$, $V_{o1,b}(n)$ includes the information of all its intermediate states.

Now, we continue iterating equation (5.29) to remove all the intermediate terms $V_{o1,b}(i)$, ($i =$

$0, 1, \dots, n - 1$) and we obtain:

$$\begin{aligned}
V_{o1,b}(n) = & \sum_{m=1}^n p_1(n-m) f_{b1,i}(m) V_{i1}(m) \\
& + V_{of1} \sum_{m=1}^n p_1(n-m) f_{b1,o}(m) + V_{of1} \sum_{m=1}^n p_1(n-m) f_{b1,a}(m) \sum_{i=0}^{m-1} f_{a1,o}(i+1) \prod_{j=i+1}^{m-1} f_{a1,a}(j+1) \\
& + V_{ref} \sum_{m=1}^n p_1(n-m) f_{b1,s}(m) S(m) + V_{ref} \sum_{m=1}^{n-1} f_{a1,s}(m+1) S(m) \sum_{i=m}^n p_1(n-i) f_{b1,a}(i) \prod_{j=i+1}^{m-1} f_{a1,a}(j+1) \\
& + p_1(n) V_{o1,b}(0) + V_{o1,a}(0) \sum_{m=1}^n p_1(n-m) f_{b1,a}(m) \prod_{i=0}^{m-1} f_{a1,a}(i+1)
\end{aligned} \tag{5.30}$$

where

$$p_1(k) = \begin{cases} 1 & k = 0 \\ f_{a1,b}(k) f_{b1,a}(k) & k = 1 \\ f_{a1,b}(k-1) f_{b1,a}(1) p_1(k-1) + f_{a1,b}(k-1) \sum_{i=2}^k f_{b1,a}(i) p_1(k-i) \prod_{j=0}^{i-2} f_{a1,a}(k-j) & k > 1 \end{cases} \tag{5.31}$$

Until now, we get the complete expression of $V_{o1,b}$. Equation (5.30) shows the contributions of the inputs, offset, modulator outputs and its initial states ($V_{o1,a}(0)$ and $V_{o1,b}(0)$) to the output of the first integrator at cycle n . If a conversion includes M clock cycles, the outputs of the first integrator can be expressed in a matrix form by considering $V_{o1,a}(0) = V_{o1,b}(0) = 0$:

$$\mathbf{v}_1^T = STF1_i \cdot \mathbf{v}_{i1}^T + V_{of1} \cdot \mathbf{stf1}_o^T + STF1_s \cdot \mathbf{s}^T \tag{5.32}$$

with:

$$\mathbf{v}_1 = [V_1(1), V_1(2), \dots, V_1(M)] \tag{5.33}$$

$$\mathbf{v}_{i1} = \mathbf{x} - V_{ref} \cdot \mathbf{s} \tag{5.34}$$

where $\mathbf{x} = [X, X, \dots, X]$ with M elements and:

$$\mathbf{s} = [S(1), S(2), \dots, S(M)] \tag{5.35}$$

The matrix of impulse responses of \mathbf{v}_{i1} , \mathbf{s} and V_{of1} are respectively:

$$STF1_i = \begin{bmatrix} a_{1,1} \cdots 0 \cdots 0 \cdots 0 \\ \vdots \cdots \vdots \cdots 0 \cdots 0 \\ a_{n,1} \cdots a_{n,m} \cdots a_{n,n} \cdots 0 \\ \vdots \cdots \vdots \cdots \vdots \cdots \vdots \\ a_{M,1} \cdots a_{M,m} \cdots \cdots \cdots a_{M,M} \end{bmatrix} \quad (5.36)$$

$$STF1_s = \begin{bmatrix} b_{1,1} \cdots 0 \cdots 0 \cdots 0 \\ \vdots \cdots \vdots \cdots 0 \cdots 0 \\ b_{n,1} \cdots b_{n,m} \cdots b_{n,n} \cdots 0 \\ \vdots \cdots \vdots \cdots \vdots \cdots \vdots \\ b_{M,1} \cdots b_{M,m} \cdots \cdots \cdots b_{M,M} \end{bmatrix} \quad (5.37)$$

$$\mathbf{stf1}_o = [c_1, \cdots, c_n, \cdots, c_M] \quad (5.38)$$

where the coefficients $a_{n,m}$, $b_{n,m}$, c_n can be calculated according to equation (5.29) by taking corresponding n and m . Regardless of the second integrator integrating in different phase, the impulse response is the same. Making an analogy with the first integrator, for the second one, supposing $V_{o2,b}(0) = V_{o2,a}(0) = 0$, we have:

$$\mathbf{v}_2^T = STF2_i \cdot \mathbf{v}_{i2}^T + STF2_s \cdot \mathbf{s}^T + V_{of2} \cdot \mathbf{stf2}_o^T \quad (5.39)$$

where $\mathbf{v}_2 = [V_2(1), V_2(2), \dots, V_2(M)]$ and $\mathbf{v}_{i2} = [V_1(1), V_1(2), \dots, V_1(M)]$. Combining equation (5.32) and equation (5.40), after M cycles, the value of the second integrator can be expressed as:

$$\begin{aligned} V_2(M) = & STF2_i(M) \cdot STF1_i \cdot \mathbf{v}_{i1}^T + STF2_i(M) \cdot STF1_s \cdot \mathbf{s}^T + STF2_s(M) \cdot \mathbf{s}^T \\ & + V_{of1} \cdot STF2_i(M) \cdot \mathbf{stf1}_o^T + V_{of2} \cdot \mathbf{stf2}_o(M) \end{aligned} \quad (5.40)$$

Combining equation (5.36), equation (5.38), equation (5.40) and equation (5.34) we obtain:

$$\begin{aligned}
X &= \frac{(STF2_i(M) \cdot STF1_i + STF2_i(M) \cdot STF1_s + STF2_s(M)) \cdot \mathbf{s}^T}{K_2} \\
&+ \frac{V_{of1} \cdot STF2_i(M) \cdot \mathbf{stf1}_o^T + V_{of2} \cdot \mathbf{stf2}_o(M)}{K_2} \\
&= \frac{V_2(M)}{K_2}
\end{aligned} \tag{5.41}$$

where

$$K_2 = \sum_{m=1}^M STF2_i(M) \cdot (STF1_i(m)^T)^T \tag{5.42}$$

In this case, the quantization error becomes $E_{q3} = \frac{V_2(M)}{K_2}$. A FIR filter with following transfer function can be use to reconstruct the modulator input, supposing we know exactly all parameters of the modulator:

$$\begin{aligned}
F_{ac}(z) &= \frac{(STF2_i(M) \cdot STF1_i + STF2_i(M) \cdot STF1_s + STF2_s(M)) \cdot \mathbf{z}^T}{K_2} \\
&+ \frac{V_{of1} \cdot STF2_i(M) \cdot \mathbf{stf1}_o^T + V_{of2} \cdot \mathbf{stf2}_o(M)}{K_2}
\end{aligned} \tag{5.43}$$

where $\mathbf{z} = [1, z^{-1}, \dots, z^{1-M}]$. We call this filter as actual filter.

5.5.2 Use of actual filter as the reconstruction filter for the non-ideal modulator

In the simulation, we use the $\Sigma\Delta$ modulator model composed of two non-ideal integrators as illustrated in Chapter 4. Since the correction performances are very sensitive to C_p , this time, we make $V_{of1} = V_{of2} = 0$, $C_{o1} = C_{o2} = 0$ and $A = 10^6$ so that we can focus on the impacts of C_{p1} and C_{p2} on the $\Sigma\Delta$ modulator. We sweep either the value of C_{p1} or C_{p2} , keeping the other value to 0. Fig. 5.21 shows ER evolution of modulators with different value of C_p , ($i = 1, 2$). For each configuration, the modulator input is reconstructed respectively using theoretical filter and actual filter. As A is great, knowing C_{p1} and C_{p2} , we can consider that we know exactly the parameters of defects. According to Fig. 5.21, we found that the performance of the modulator is very sensitive to the $C_{p,i}$. Apart from this, we find that even though using the actual filter, we cannot get a much better resolution than using the theoretical filter. However, ER can be always improved by applying correction, here it is the optimal filter correction.

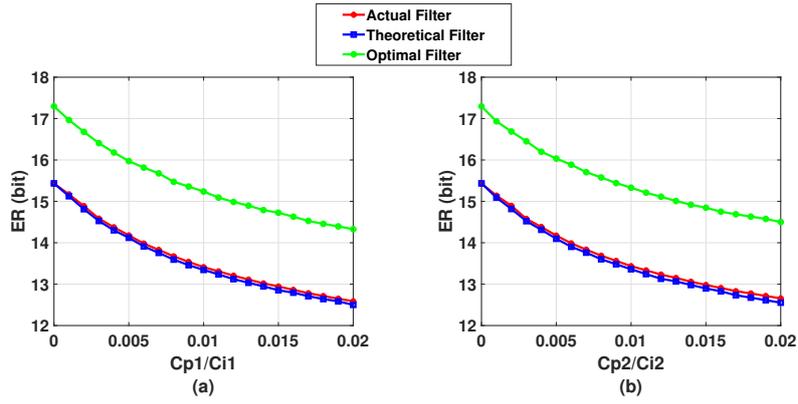


Figure 5.21: ER in the function of C_p applying different reconstruction filters in the simulation of the proposed $\Sigma\Delta$ modulator model

5.5.3 Error analysis

With C_p , the transfer function of the modulator is changed. As illustrated in Chapter 4, we already knew that there is a degradation if the theoretical filter is employed to reconstruct modulator input for a non-ideal modulator. However, the principle of sigma-delta modulation is to approximate modulator input by minimizing the residue. According to equation (3.6), for an ideal modulator, residue is divided by K_1 where $K_1 = \frac{2}{a_1 a_2 M(M-1)}$ after the conversion. While with $C_{p,i}$, it is only divided by K_2 , according to equation (5.41). If $C_{p,i}$ is large, we have $K_2 \ll K_1$, which means that the residue is not minimized sufficiently to achieve the desired resolution. It is probable that $\frac{V_2(M)}{K_1} + \epsilon \approx \frac{V_2(M)}{K_2}$, where ϵ is defined in equation (4.3). So even though we know exactly the transfer function of the modulator (actual filter), we cannot improve significantly ER . For this reason, correction is mandatory.

5.6 Conclusion

In this chapter, we propose a new circuit-level model of inverter-based SC integrators with parasitic capacitors which can well explicate the behaviors of the $\Sigma\Delta$ modulator in post-layout simulation. Hence the $\Sigma\Delta$ modulator is no longer a fully black-box to us.

Based on the new integrator model, we can further study the limit of the correction methods proposed in Chapter 3 as well as the existing method, with different level of defects. According to the simulation results, we discover that performance of correction methods are sensitive to C_p . Thus employing inverter-based integrator, it is necessary to pay more attention in design to avoid C_p . We can also study noise resistance with the help of the proposed model. To maintain ER over 14 bits, in two-step mode, noise level is supposed to be lower than 12 bits. In addition to it, we are

able to model and simulate the principle defects of the $\Sigma\Delta$ modulator with a model on MATLAB.
The simulation of the model on MATLAB takes less time than the post-layout simulations.

Chapter 6

Measurement

The design of the analog part of the $\Sigma\Delta$ modulator had been finished by Pierre Bisiaux at the beginning of my PhD career. And during the first six months of my PhD, I completed the digital part. Then with the help of a technician in our group, the layout of the $\Sigma\Delta$ modulator had been finished and we named the final prototype chip as Pieretta. It was sent to fabrication at the end of the first year of my PhD and we received it during the second year. In this situation, I was able to complete measurement and confront the theory and the real chip during the third year.

The first measurements show that our circuit is interfered by noise at extremely high level, for both analog and digital signals. On one hand, the noise in digital signals disturbed sampling as the sampling at each clock cycle was enabled by a pulses while the undesired peaks in this signal were mistaken for the enable signals. On the other hand, the noise degraded the resolution of the $\Sigma\Delta$ modulator. For this reason, the $\Sigma\Delta$ modulators only achieved 8-bit resolution. In this case, it was impossible to study the correction method performance since the correction methods serve to eliminate finer errors.

In order to eliminate the impact of the noise on sampling, we developed an asynchronous sampling method whose details will be presented later. And to eliminate the impact of the noise on modulator resolution, we performed several conversions for the same input value and took the average. In this way, we rescued the circuit as much as possible.

6.1 Package of Pieretta

Pieretta chip has been fabricated with the technology of XFAB 180 nm. As shown in Fig. [6.2](#), five $\Sigma\Delta$ modulators, whose layout is shown in Fig. [6.1](#), are integrated in the core of Pieretta. In Fig. [6.1](#), the wires for the digital feedback are placed far away from analog circuit so that there is

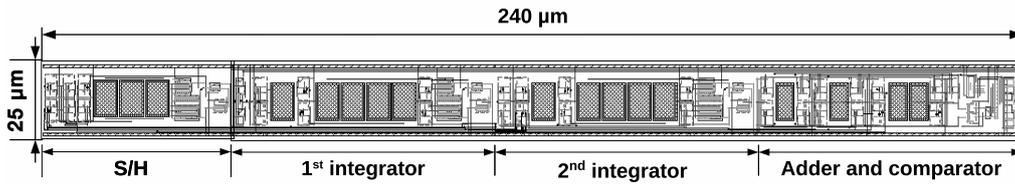


Figure 6.1: Layout of the IΣΔ modulator

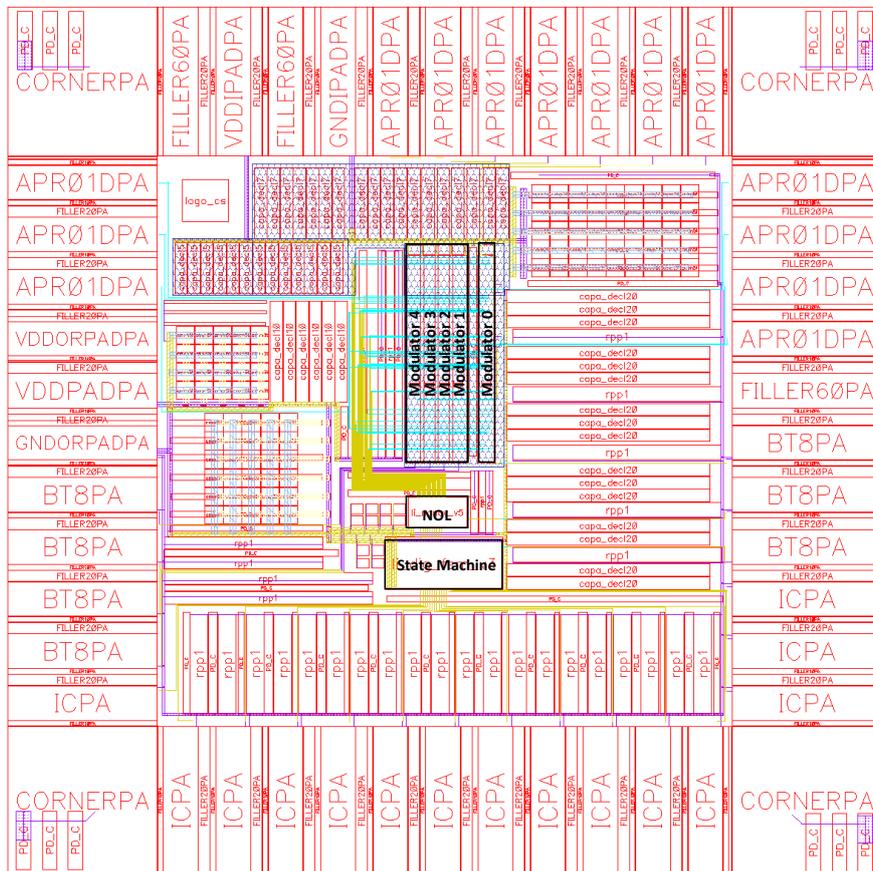


Figure 6.2: Layout of the core of Pieretta

less coupling interference. To be noticed that modulator 1 to modulator 4 locate closely side by side in order to simulate column parallel ADCs. While modulator 0 which is placed separately to others is special one. The pins of the input and the residue of modulator 0 are reserved so that we can observe them. It may come in handy in test. All these modulators share a state machine

Table 6.1: Descriptions of conversion modes with different configurations

Pin Number	Name	Descriptions
25	<i>BS0</i>	Output of modulator 0
2 – 5	<i>BS1</i>	Outputs of modulator 1-4
32	<i>residue</i>	Residue of modulator 0
33	<i>vin_modulator</i>	Output of S/H of modulator 0
34 – 38	<i>vin_pixel0 – 4</i>	Inputs of modulator 0-4
23	<i>step1</i>	Indication of the first step
33	<i>Clock_out</i>	Indication of the moment to sample modulator output
8	<i>start</i>	Enable conversion
9	<i>rst_sys</i>	Global Reset signal
10	<i>mode_infi</i>	Infinite mode
11	<i>mode_prog</i>	Programmable mode
12, 13	<i>NB_step0 – 1</i>	Numbers of steps
14, 15	<i>NB_samp0 – 1</i>	Length of sampling period
16 – 22	<i>OSR0 – 6</i>	Numbers of clock cycles in each step
27	<i>IBP_TOP</i>	Bias current of PMOS
28	<i>IBN_TOP</i>	Bias current of NMOS
29	<i>IBP_TH</i>	Bias current of PMOS
30	<i>IBN_TH</i>	Bias current of NMOS
39	<i>GNDA</i>	Analog ground
41	<i>VDDA</i>	Digital VDD
44	<i>VREFP</i>	Reference Voltage of positive feedback
45	<i>VREFN</i>	Reference Voltage of negative feedback
46	<i>VCM</i>	Common mode Voltage
47	<i>VDDR</i>	VDD
48	<i>VDD!</i>	VDD
1	<i>GND!</i>	Ground

and the non-overlap block. The packaged chip has a die area of 1.5mm x 1.5mm. The differential output driver is placed near the pads, and the power is supported from the dedicated supply and ground lines. The spare chip area excluding the core of Pieretta is filled with decoupling capacitors for density equilibrium. There are total 48 I/O pins using DIP (dual in-line package). The pin description is summarized in Table 6.1.

6.2 Equipment of Test-bench

Global schematic of the test-bench is shown in Fig. 6.3. The test-bench includes a PC, a signal generator, a source meter, an Arduino, a logical analyzer with an oscilloscope and a PCB for

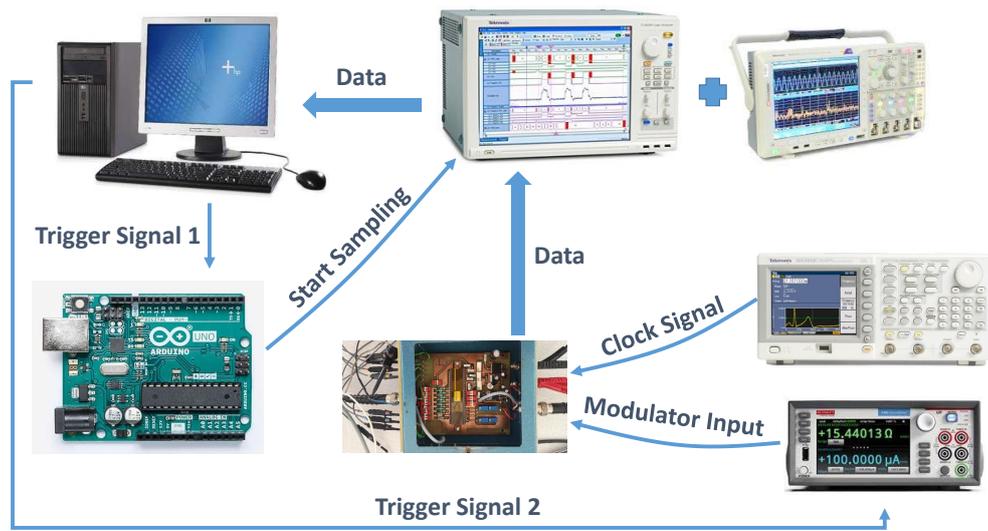


Figure 6.3: Schematic of test-bench

Pieretta. The signal generator generates an external clock signal to Pieretta while the source meter generates the modulator inputs. The PC sends trigger signals to Arduino and controls the values of the source meter. It also handles data post-processing. Once Arduino receives the trigger signal from PC, it sends a signal to the logical analyzer to enable sampling. During the conversions, the logical analyzer samples the modulator outputs and send them back to the PC for post-processing. The waveform of the signals can be observed on the oscilloscope if necessary.

In order to reduce interference of noise, the PCB for Pieretta is designed to be as simple as possible. The modulator parameters are configured by changing the states of switches. To be noticed that, we use batteries as power supply instead of an industrial voltage source meter because we find that the voltage source also brings about distribution as it is powered with alternating current.

6.3 Asynchronous Sampling

We intended to sample the modulator output bits in a synchronous way with the help of the signal *clk_out* generated by the modulator. In this way, there is no need of Arduino which sends trigger signal to the logical analyzer. For each bit, the logical analyzer samples at the rising edge of *clk_out* as the modulator output is well established. With the help of *step1*, each conversion can be recognized easily. The timing diagram is shown in Fig. 6.4. However, these signals are polluted by cross-talking between the digital signals so there are so many undesired glitches in *clk_out* that the logical analyzer treats the glitches as the rising edge of *clk_out* mistakenly. As a results, the

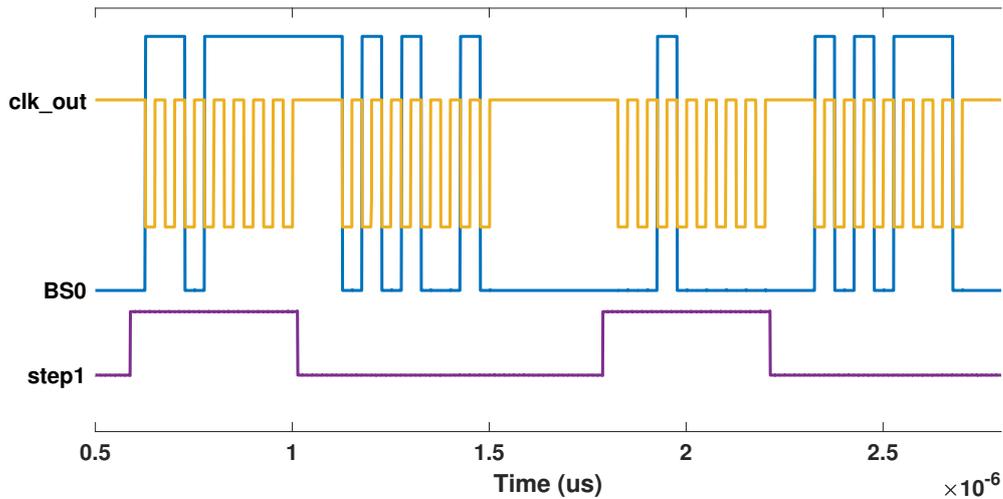


Figure 6.4: Timing diagram of clk_out and the modulator output

logical analyzer samples at wrong moment and the modulator output bits obtained are incorrect.

To solve this problem, we decide to sample in an asynchronous way. That means the modulator output bits as well as clk_out are sampled by the logical analyzer which is commanded by a intern signal clk . The frequency of clk is at least ten times greater than that of clk_out . Thus clk_out and S are converted to a series of '1' or '0', we define these signal respectively as clk_out' and S' . The waveform the signals for sampling are shown in Fig. 6.5. Then we need to clean clk_out' by inverting the bits that are different to both its previous and later bits. Now we smooth out the glitches in clk_out' and we can re-synchronize the modulator output bits by detecting the indexes of rising edges in clk_out' .

In fact, the glitches not only exist in clk_out but also in the modulator outputs. Therefore, we create a voting mechanism to decide the value of S . The modulator output bits at rising edges of clk_out' and its neighboring bits are voting members possessing the same weights. If the average is greater than 0.5, we make S equal to 1, else S equal to 0. Fig. 6.6 shows the re-synchronization and the voting mechanism.

6.4 Measurement Results

6.4.1 Interference of Noise and Primary Measurement Results

We observe strong interference of noise in the circuit which is mainly reflected in three aspects.

First, there are many glitches in the output digital signals. This noise in digital signals not only

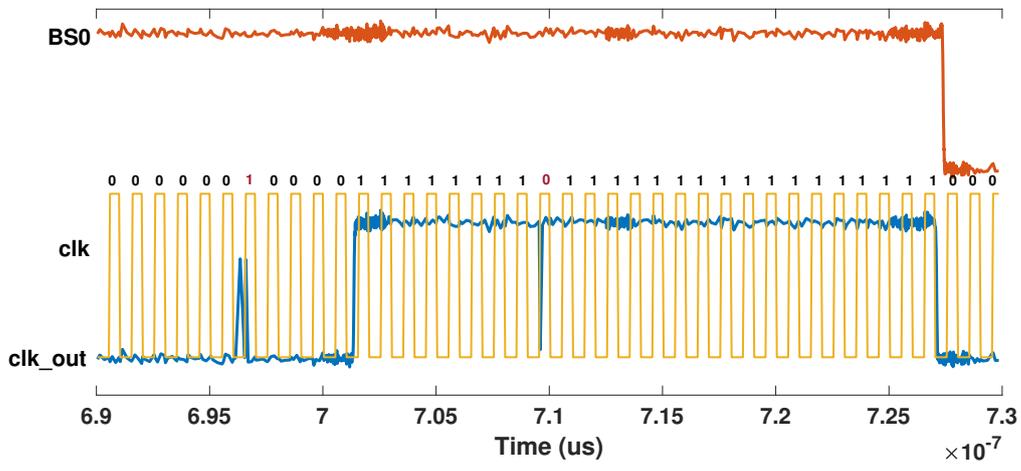


Figure 6.5: Timing diagram of signals in asynchronous sampling mode

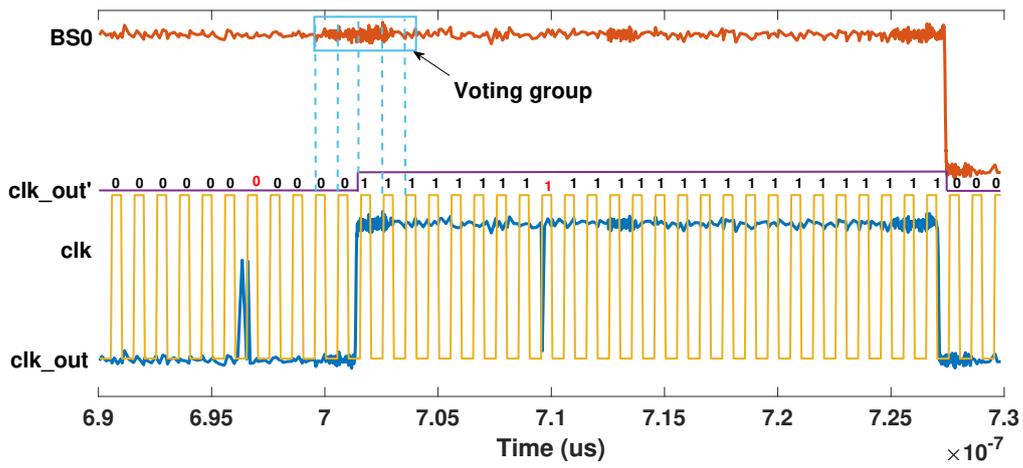


Figure 6.6: Generation of clk_out' and the voting mechanism for deciding the values of BS

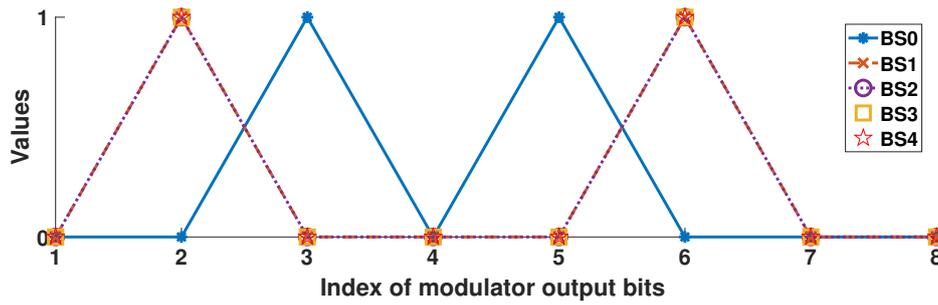


Figure 6.7: Comparison between the first 8 MSBs in outputs of different modulators for the same input

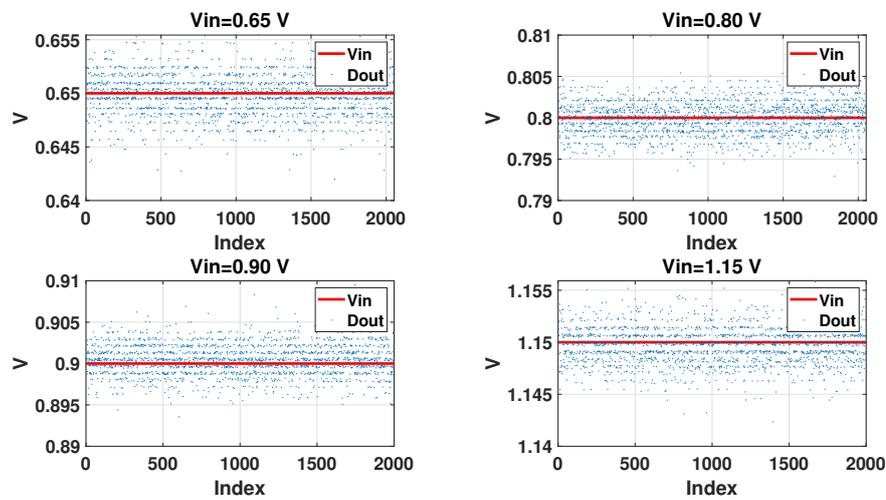


Figure 6.8: Comparison between the modulator outputs and the modulator inputs

has an impact on sampling as illustrated in Section 6.3, but also interferes with internal analog signals.

Second, the output bit sequences of each modulator are different when their inputs are the same. As shown in Fig. 6.7, even from the second bit, the outputs start to be different. This phenomenon usually happens to all modulators.

Third, even if constant inputs are applied, in each time, the outputs of a modulator vary. Fig. 6.8 shows comparison between certain output values of modulator m0 which are reconstructed with the correction of gain and offset and the inputs. Conversions are executed in two-step mode with $M_1 = 48$, $M_2 = 24$ and the frequency of clock is 20 MHz.

Due to the noise, ER is largely degraded according to a test with the same conditions. A complete test is done where the input varies from 0.60 to 1.20 with a step of 0.002 V. Conversions are performed over two steps with $M = 72$, $M_1 = 48$. We obtain $ER = 8.5$ after the optimal filter in plan B. The form of conversion errors is shown in Fig. 6.9. The degradation is caused by noise.

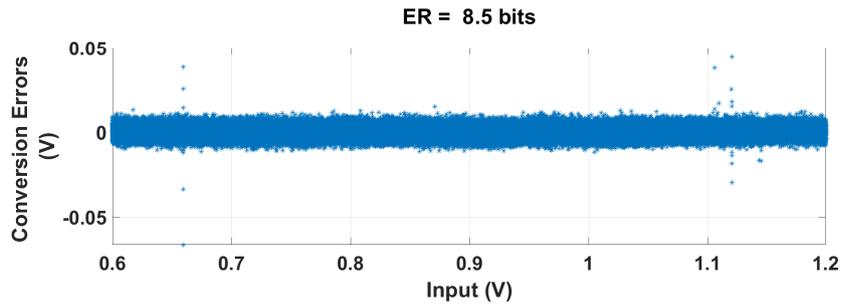


Figure 6.9: Original conversion errors of the real circuits

6.4.2 Filtered Measurement Results

As noise level is much higher than the target LSB error level, to study the correction performances in the real circuit, we have to filter out noise. 2048 samples are taken for the same input value. We first perform a pre-processing to eliminate the extreme samples. We reconstruct the modulator input with the optimal filter in plan B then we remove the samples whose absolute value of conversion error is larger than 0.02 V.

Then we calculate the average error of the rest of valid samples for each input value. As shown in Fig. 6.10 the filter is positioned right after the modulator. Therefore, the modulator outputs to be corrected is filtered. In this way, we can verify performance of correction methods eliminating the effect of noise.

Fig. 6.11 (a) shows the form of conversion errors with the optimal filter in plan A. With the optimal filter, resolution is still far away from the target. After canceling most part of noise in the modulator output, ER increase significantly. Fig. 6.11 (b) shows the form of conversion errors corrected with the pattern-correcting filter of 3-bit package for the first 3 MSBs. With the pattern-correcting filter, ER can be further improved. While Fig. 6.11 (c) shows that with the simplified filter for the first step. The ER in (c) has a little degradation comparing to (a). Fig. 6.11 (d) is for the hybrid filter correction of 3-bit package for the first 3 MSBs. It is close to the ER in (a), however it requires less coefficients.

Even though there is a degradation in all ERs in the real circuit comparing to those in post-layout simulation, the proposed correction methods are proven to be efficacious. The degradation is maybe still caused by residual noise. Or there is another possibility that the parasitic capacitors C_p circuit are much larger than those in the post-layout simulation.

Fig. 6.12 shows the results of different correction methods and simulation of ideal model, post-layout simulation and in the measurement after filtering in two-step mode. Even though compared



Figure 6.10: Filter before correction

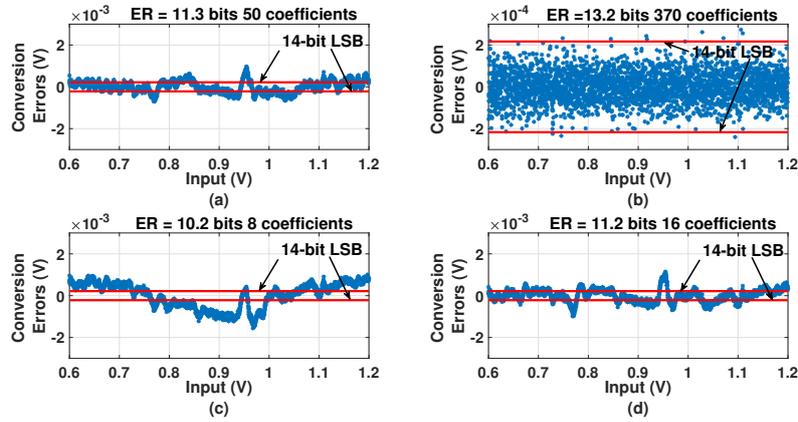


Figure 6.11: Conversion errors after correction, choice 1: (a) optimal filter in plan A; (b) pattern-correcting filter in plan A, $m = 3$; (c) simplified filter; (d) hybrid filter, $m = 3$, $m_{bit} = 3$;

to post-layout simulation, there is shift in ER in measurement for all scenes, the tendency is almost similar:

- a. The pattern-correcting filter can further improve ER than the optimal filter.
- b. The simplified filter can offer ER close to that applying the optimal filter, however the simplified filter requires less coefficients. It is its advantage compared to the optimal filter.
- c. Compared to the simplified filter, the hybrid filter can still improve resolution by simply correcting the first three MSB. However, the improvement is larger in measurement.

6.5 Conclusion

The real circuit suffered from the noise of abnormal high power exceptionally. The noise caused problem in both sampling and post-processing in the test. On the one hand, In order to obtain correct samples, we developed an asynchronous sampling method and applied it to the logic analyzer. In this way, we were able to eliminate the interference of noise in the signal controlling sampling. On the other hand, in the post-processing, due to the powerful noise, the effect of corrections is covered by noise. For verifying the performance of correction methods, we thought of filtering out these noises in post-processing on Matlab. Measurement results shows that with the filtered modulator outputs, regardless of a global degradation in ERs , the optimal filter and all proposed correction methods can indeed improve resolution in real circuit. Despite of a global

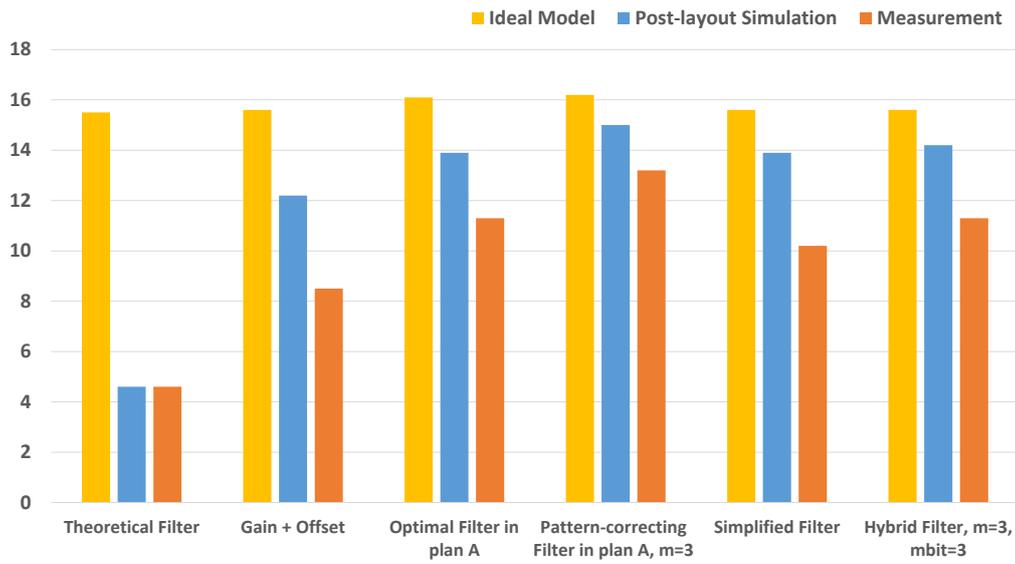


Figure 6.12: Tendency of ER in two-step simulations.

degradation in ER for all correction methods, the tendency of correction results in the real circuit is coherent to the tendency in the schematic simulation and the post-layout simulation as well.

Chapter 7

Conclusion

This research presents the attempt to develop a calibration technique for our $\Sigma\Delta$ ADC using inverter-based SC integrators, so that it achieves the target resolution. Since the only information accessible from the $\Sigma\Delta$ ADC is the modulator output bit stream, we first aim at finding an algorithm to correct the modulator output based on output bit stream. In this step, we don't think about parameter identification.

The classical correction $\Sigma\Delta$ ADC is the optimal filter correction which corrects each bit individually. It is a linear function of the modulator output bits. However, the conversion error isn't completely a linear function of the modulator output bits. Considering the ADC as a black-box, we propose the pattern-correcting filter correction which detects the combination of every three consecutive modulator output bits. Since it takes into account the sequence of bits, we can consider that it is a non-linear correction method. Even though the pattern-correcting filter can improve equivalent resolution largely, it requires much more coefficients that increases complexity. To make it implementable, we simplify the optimal filter by approximating its impulse response by a IIF in order to reduce the number of coefficients. Then we combine the pattern-correcting filter to the simplified filter to enhance equivalent resolution. We call this correction method as hybrid filter correction. The pattern-correcting filter only corrects the first three MSBs. Thus, it doesn't need thousands of coefficients. With the hybrid filter correction, the ADC finally achieve 14 bits with an acceptable number of coefficients.

Observing certain abnormal intern signals of the $\Sigma\Delta$ modulator in post-layout simulation, we speculate that there are some exceptional parasitic capacitors. One is a capacitor crossing the input and the output of inverter-based amplifier. This capacitor C_p brings memory error to the output of inverter-based integrator. Another is a capacitor between the amplifier input and the modulator output. It leads to inter-modulation to integrators. Based on these parasitic capacitors,

a new model of inverter-based SC integrator is proposed. This model can well explicate the abnormal behaviors of those intern signals in our $I\Sigma\Delta$ modulator. With these parasitic capacitors, the degradation of our modulator becomes reasonable. Although we cannot correct the errors brought about by the parasitic capacitors even though we know its values, this error model helps us to study the limit of corrections methods and their noise resistance. It saves time to simulate with the error model instead of doing Monte Carlo simulations in post-layout level.

The measurement results show an exceptional strong noise interference in the circuit. The level of noise is so great that it covers the improvement of corrections. In order to verify the effect of correction methods, the modulator outputs are filtered by taking averages before corrections. In this way, the effect of corrections is revealed. In spite of a global degradation in equivalent resolution for all correction methods, we obtain the same tendency. It indicates that the proposed correction methods are efficacious.

Appendix A

Study of Convergence in Parameter Estimation

As presented, calibration includes two aspects: correction algorithm and parameter identification. We have defined several digital reconstruction filters and their coefficients have been estimated by researching least-mean-square of conversion errors based on the recorded data set of post-layout simulation. Since it is a off-line calibration, parameter identification should be performed before normal conversions. And once the parameters (coefficients of the digital reconstruction filters) are estimated, they cannot change during the normal conversions. Therefore, The stability of the parameter identification is very important. However, As we all know, the values of estimated coefficients may vary if we use different data set for the estimation. The objective of this research is to study the convergence of estimated coefficients and to its impact to the correction results.

We applied a set of inputs which distribute uniformly between $0.8V_{ref}$ with the step of certain value as the data set in coefficient estimation. By decreasing the value of the step of inputs, we can increase the size of the data set applied for coefficient estimation. Then we applied another set of 2^{12} inputs distributing randomly between $0.8V_{ref}$ for testing. For collecting a huge data set, we choose the two-step $\Sigma\Delta$ ADC model proposed in Chapter 5 with $M_1 = 48$, $M_2 = 24$ instead of post-layout simulation in order to save time. At the first time, simulations without noise have been done on Matlab. We increased the number of samples used for coefficients estimation step by step and recorded the estimated values of coefficients in each time. We observed the differences of the values estimated of the same coefficient in adjacent simulation, if the error is stable within an acceptable range, we can think this coefficient value converges. For seeing the convergence in the environment of noise, the simulations of the same model with noise whose amplitude is

$10^{-4}V$ were done as illustrated in Chapter 5, it is the acceptable maximum noise level for 14 bits in two-step mode.

Fig. A.1 shows the differences of estimated values of the coefficients with the increasing numbers of samples used for coefficient estimation, in the correction method correcting gain and offset. The red lines indicate the LSB error to code this coefficient. As illustrated in Chapter 4, 15 bits are enough to code the coefficients of the filter correcting gain and offset. We find that whether there is noise or not, the estimated values of coefficients converge to certain values with the increasing size data. Besides, ER is around 9.8 bits.

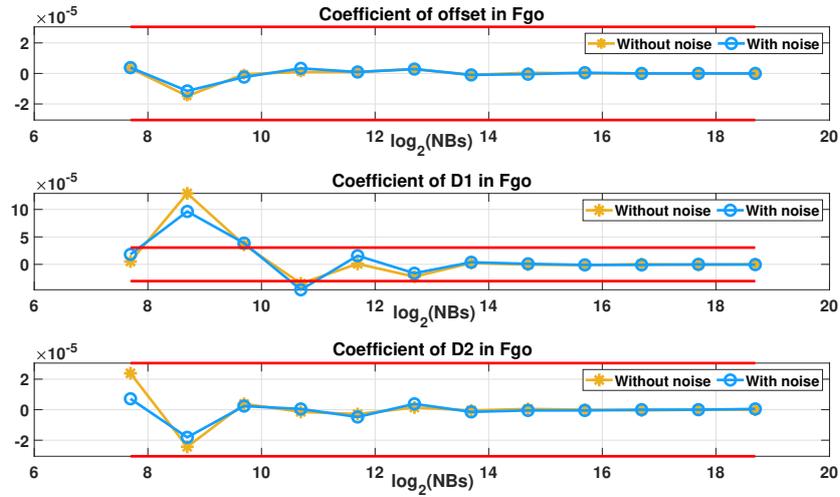


Figure A.1: Differences of the values estimated of the coefficients correcting gain and offset in the function of the numbers of samples used for coefficient estimation.

Fig. A.2 show the differences of the value estimated of the coefficients for offset and D_2 in the optimal filter of plan A. For all the coefficients of the optimal filter, 18 bits are required to code the coefficients. Similar to the situation in the previous correction method, the estimated values of coefficients converge too to certain values at the end. Fig. A.2 show the differences of the coefficients values estimated of the first three MSBs of the modulator outputs BS . They also converge. Regardless of the numbers of samples, ER finally stabilized at 14 bits.

For the simplified filter, there are two types of coefficients: the coefficients in FIR, $D_{go,2}$ and O_{si} , and those for the poles and zeros in the second-order IIR F_{si} . To code the former, 18 bits are needed and as shown in Fig. A.4, $D_{go,2}$ and O_{si} converge if the number of samples used for estimating coefficients is large enough. To code the coefficients (p_0 to p_2) for the zeros in F_{si} , at least 16 bits are needed while for those of poles (q_0 to q_2), 18 bits are necessary. As shown in Fig. A.5, most of coefficients in F_{si} converge at the end, except p_1 which almost converges. For this correction methods, ER tends to 13.1 bits. On the basis of the simplified filter, the hybrid filter needs additional 8 coefficients to correct the pattern of the first three MSBs of the modulator

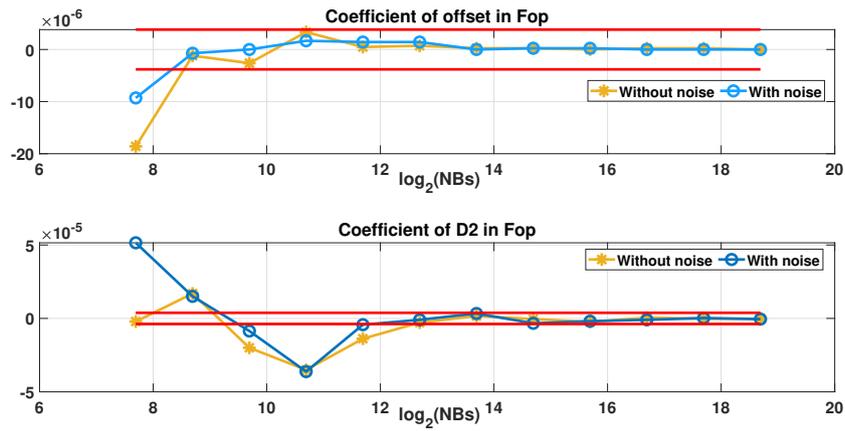


Figure A.2: Differences of the values estimated of the coefficients correcting offset and D_2 in the optimal in the function of the numbers of samples used for coefficient estimation.

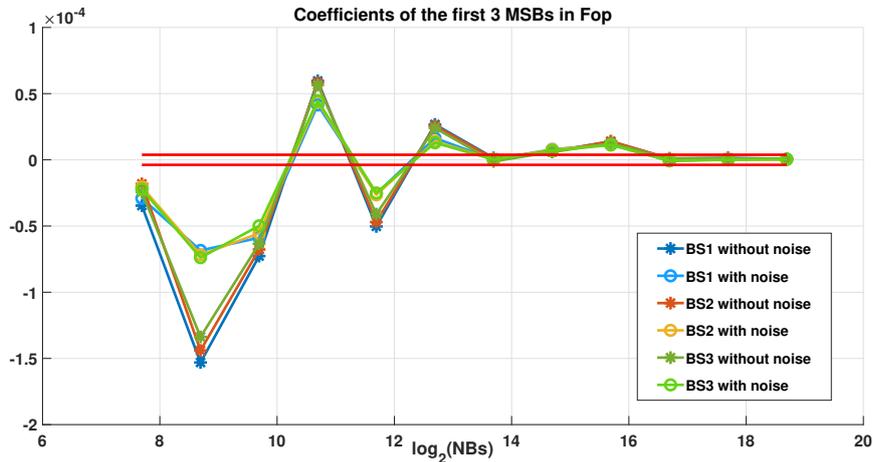


Figure A.3: Differences of the values estimated of the coefficients of the first three MSBs in the optimal in the function of the numbers of samples used for coefficient estimation.

outputs. According to the simulations, only the coefficients for 110 and 111 are stable, in fact, their values are 0. While the differences of coefficients for rest of patterns vary between three values: 0 and the values near 1. Fig. [A.6](#) shows the differences of the coefficients of pattern 000 and 001. Although the curves of two patterns coincide, if we look at directly the values of differences, we will find that they are not equal. We didn't plot the coefficients of other patterns because the all curves coincident. It means that the values of these coefficients in fact oscillate between certain value. Even though the coefficients don't converge, ER can be always keep around 14 bits. In summary, for all correction filters, the coefficients correcting offset and D_2 are easy to converge. If a coefficient converges when we increase the number of samples applied for estimation, it will still converge regardless of noise. However, coefficients do not necessarily need to converge. There

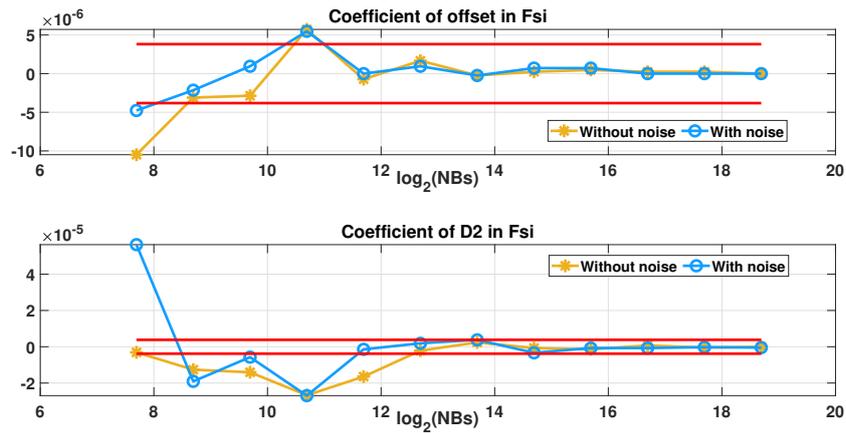


Figure A.4: Differences of the values estimated of the coefficients correcting offset and D_2 in the simplified filter in the function of the numbers of samples used for coefficient estimation.

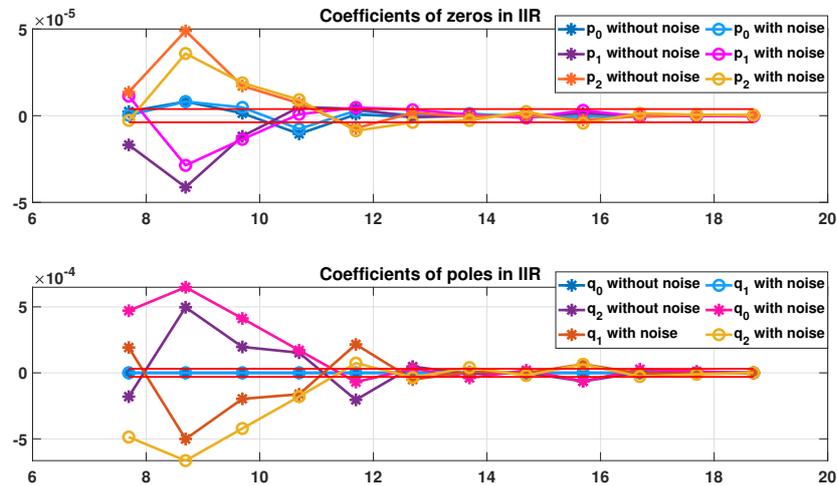


Figure A.5: Differences of the values estimated of the coefficients in IIR filter in the simplified filter in the function of the numbers of samples used for coefficient estimation.

are several coefficient combinations working well. However, for each combination, the estimation accuracy of the coefficients values is important. Combining with the illustration in Chapter 4, the key to keep correction performance is to use enough bits to code each coefficients.

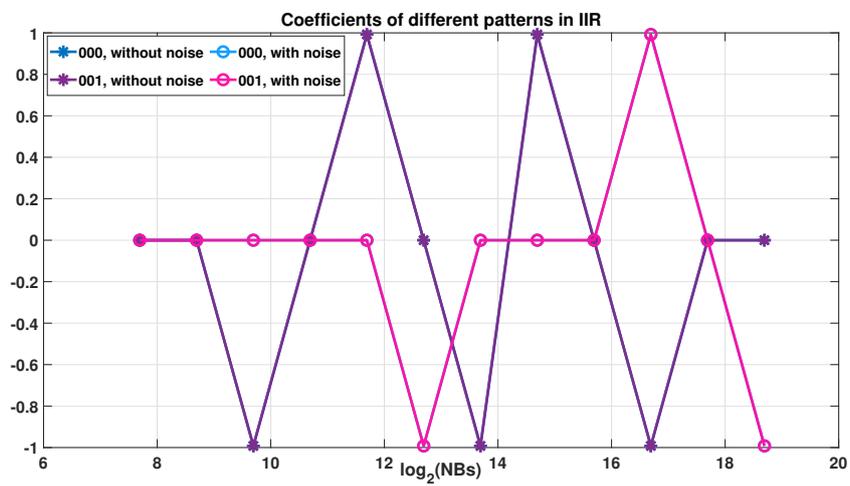


Figure A.6: Differences of the values estimated of the coefficients of certain patterns in the hybrid filter in the function of the numbers of samples used for coefficient estimation.

Bibliography

- [1] Sonia Vargas Sierra, Elisenda Roca Moreno, and Gustavo Liñán Cembrano. Pixel design and evaluation in cmos image sensor technology. In *XXIV Conference on Design of Circuits and Integrated Systems (2009)*, p 1-6, 2009.
- [2] Fayçal Saffih. Foveated sampling architectures for cmos image sensors. 2005.
- [3] Niladri Pratap Maity and Reshmi Das. Design of low noise high fill-factor cmos active pixel image sensor with correlated double sampling technique.
- [4] Yoshikazu Nitta, Yoshinori Muramatsu, Kiyotaka Amano, Takayuki Toyama, K Mishina, Atsushi Suzuki, Tadayuki Taura, Akihiko Kato, Masaru Kikuchi, Yukihiro Yasui, et al. High-speed digital double sampling with analog cds on column parallel adc architecture for low-noise active pixel sensor. In *2006 IEEE International Solid State Circuits Conference-Digest of Technical Papers*, pages 2024–2031. IEEE, 2006.
- [5] Yue Chen, Yang Xu, Adri J Mierop, and Albert JP Theuwissen. Column-parallel digital correlated multiple sampling for low-noise cmos image sensors. *IEEE Sensors Journal*, 12(4): 793–799, 2011.
- [6] Takayuki Toyama, Koji Mishina, Hiroyuki Tsuchiya, Tatsuya Ichikawa, Hiroyuki Iwaki, Yuji Gendai, Hirotaka Murakami, Kenichi Takamiya, Hiroshi Shiroshita, Yoshinori Muramatsu, et al. A 17.7 mpixel 120fps cmos image sensor with 34.8 gb/s readout. In *2011 IEEE International Solid-State Circuits Conference*, pages 420–422. IEEE, 2011.
- [7] Mitsuhiro Mase, Shoji Kawahito, Masaaki Sasaki, Yasuo Wakamori, and Masanori Furuta. A wide dynamic range cmos image sensor with multiple exposure-time signal outputs and 12-bit column-parallel cyclic a/d converters. *IEEE Journal of Solid-State Circuits*, 40(12): 2787–2795, 2005.
- [8] Jong-Ho Park, Satoshi Aoyama, Takashi Watanabe, Tomoyuki Akahori, Tomohiko Kosugi, Keigo Isobe, Yuichi Kaneko, Zheng Liu, Kazuki Muramatsu, Takeshi Matsuyama, et al. A

- 0.1 e-vertical fpn 4.7 e-read noise 71db dr cmos image sensor with 13b column-parallel single-ended cyclic adcs. In *2009 IEEE International Solid-State Circuits Conference-Digest of Technical Papers*, pages 268–269. IEEE, 2009.
- [9] Li Quanliang, Liu Liyuan, Han Ye, Cao Zhongxiang, and Wu Nanjian. A 12-bit compact column-parallel sar adc with dynamic power control technique for high-speed cmos image sensors. *Journal of Semiconductors*, 35(10):105008, 2014.
- [10] Ruoyu Xu, Wai Chiu Ng, Jie Yuan, Shouyi Yin, and Shaojun Wei. A 1/2.5 inch vga 400 fps cmos image sensor with high sensitivity for machine vision. *IEEE Journal of Solid-State Circuits*, 49(10):2342–2351, 2014.
- [11] Youngcheol Chae, Jimin Cheon, Seunghyun Lim, Minho Kwon, Kwisung Yoo, Wunki Jung, Dong-Hun Lee, Seogheon Ham, and Gunhee Han. A 2.1 m pixels, 120 frame/s cmos image sensor with column-parallel $\delta\sigma$ adc architecture. *IEEE Journal of Solid-State Circuits*, 46(1): 236–247, 2010.
- [12] Youngcheol Chae and Gunhee Han. Low voltage, low power, inverter-based switched-capacitor delta-sigma modulator. *IEEE Journal of Solid-State Circuits*, 44(2):458–472, 2009.
- [13] Biao Wang, Meng Zhang, Xu Cheng, Qi Feng, and Xiaoyang Zeng. A 1.8-v 14-bit inverter-based incremental $\sigma\delta$ adc for cmos image sensor. In *2013 IEEE 10th International Conference on ASIC*, pages 1–4. IEEE, 2013.
- [14] Toshihisa Watabe, Kazuya Kitamura, Takehide Sawamoto, Tomohiko Kosugi, Tomoyuki Akahori, Tetsuya Iida, Keigo Isobe, Takashi Watanabe, Hiroshi Shimamoto, Hiroshi Ohtake, et al. A 33mpixel 120fps cmos image sensor using 12b column-parallel pipelined cyclic adcs. In *2012 IEEE International Solid-State Circuits Conference*, pages 388–390. IEEE, 2012.
- [15] Yun Chiu, Paul R Gray, and Borivoje Nikolic. A 14-b 12-ms/s cmos pipeline adc with over 100-db sfdr. *IEEE Journal of Solid-State Circuits*, 39(12):2139–2151, 2004.
- [16] Hsin-Shu Chen, Bang-Sup Song, and Kantilal Bacrania. A 14-b 20-msamples/s cmos pipelined adc. *IEEE Journal of Solid-State Circuits*, 36(6):997–1001, 2001.
- [17] Jae-hong Kim, Wun-ki Jung, Seung-hyun Lim, Yu-jin Park, Won-ho Choi, Yun-jung Kim, Chang-eun Kang, Ji-hun Shin, Kyo-jin Choo, Won-baek Lee, et al. A 14b extended counting adc implemented in a 24mpixel aps-c cmos image sensor. In *2012 IEEE International Solid-State Circuits Conference*, pages 390–392. IEEE, 2012.

- [18] Min-Woong Seo, Takehide Sawamoto, Tomoyuki Akahori, Zheng Liu, Tetsuya Iida, Taishi Takasawa, Tomohiko Kosugi, Takashi Watanabe, Keigo Isobe, and Shoji Kawahito. A low-noise high-dynamic-range 17-b 1.3-megapixel 30-fps cmos image sensor with column-parallel two-stage folding-integration/cyclic adc. *IEEE transactions on electron devices*, 59(12):3396–3400, 2012.
- [19] Christer Jansson. A high-resolution, compact, and low-power adc suitable for array implementation in standard cmos. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 42(11):904–912, 1995.
- [20] Min-Seok Shin, Jong-Boo Kim, Yun-Rae Jo, Min-Kyu Kim, Bong-Choon Kwak, Hyeon-Cheon Seol, and Oh-Kyong Kwon. Cmos x-ray detector with column-parallel 14.3-bit extended-counting adcs. *IEEE transactions on electron devices*, 60(3):1169–1177, 2013.
- [21] Junan Lee, Himchan Park, Bongsub Song, Kiwoon Kim, Jaeha Eom, Kyunghoon Kim, and Jinwook Burm. High frame-rate vga cmos image sensor using non-memory capacitor two-step single-slope adcs. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 62(9):2147–2155, 2015.
- [22] Jaeyoung Bae, Daeyun Kim, Seokheon Ham, Youngcheol Chae, and Minkyu Song. A two-step a/d conversion and column self-calibration technique for low noise cmos image sensors. *Sensors*, 14(7):11825–11843, 2014.
- [23] M-S Shin and O-K Kwon. 14-bit two-step successive approximation adc with calibration circuit for high-resolution cmos imagers. *Electronics letters*, 47(14):790–791, 2011.
- [24] Jong-Boo Kim, Seong-Kwan Hong, and Oh-Kyong Kwon. A low-power cmos image sensor with area-efficient 14-bit two-step sa adcs using pseudomultiple sampling method. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 62(5):451–455, 2015.
- [25] Mengyun Yue, Dong Wu, and Zheyao Wang. A 15-bit two-step sigma-delta adc with embedded compression for image sensor array. In *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, pages 2038–2041. IEEE, 2013.
- [26] Chia-Hung Chen, Yi Zhang, Tao He, Patrick Y Chiang, and Gabor C Temes. A micro-power two-step incremental analog-to-digital converter. *IEEE Journal of Solid-State Circuits*, 50(8):1796–1808, 2015.
- [27] Olivier Saint-Pé, Michel Tulet, Robert Davancens, Franck Larnaudie, Pierre Magnan, Franck Corbière, Philippe Martin-Gonthier, and Pierre Belliot. grade cmos image sensors for de-

- manding space applications. In *5th International Conference on Space Optics*, volume 554, pages 775–779, 2004.
- [28] Pierre Bisiaux, Caroline Lelandais-Perrault, Anthony Kolar, Filipe Vinci dos Santos, and Philippe Benabes. A 14-bit 250 ks/s two-step inverter-based incremental σ adc for cmos image sensor in $0.18\text{-}\mu\text{m}$ technology. *Analog Integrated Circuits and Signal Processing*, 97(3):427–435, 2018.
- [29] Thomas Christen. A 15-bit 140- μw scalable-bandwidth inverter-based $\delta\sigma$ modulator for a mems microphone with digital output. *IEEE Journal of Solid-State Circuits*, 48(7):1605–1614, 2013.
- [30] Sebastian Zeller, Christian Muenker, Robert Weigel, and Thomas Ussmueller. A 0.039 mm² inverter-based 1.82 mw 68.6 db-sndr 10 mhz-bw $\sigma\delta$ -adc in 65 nm cmos using power- and area-efficient design techniques. *IEEE Journal of Solid-State Circuits*, 49(7):1548–1560, 2014.
- [31] Youngcheol Chae and Gunhee Han. A low power sigma-delta modulator using class-c inverter. In *2007 IEEE Symposium on VLSI Circuits*, pages 240–241. IEEE, 2007.
- [32] Youngcheol Chae, Inhee Lee, and Gunhee Han. A 0.7 v 36 μw 85db-dr audio $\delta\sigma$ modulator using class-c inverter. In *2008 IEEE International Solid-State Circuits Conference-Digest of Technical Papers*, pages 490–630. IEEE, 2008.
- [33] Yajunath Kaliyath and Tonse Laxminidhi. A 1.8 v 11.02 μw single-ended inverter-based ota with 113.62 db gain. In *2016 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, pages 237–241. IEEE, 2016.
- [34] Kian Ann Ng and Yong Ping Xu. 11.6 a multi-channel neural-recording amplifier system with 90db cmrr employing cmos-inverter-based otas with cmfb through supply rails in 65nm cmos. In *2015 IEEE International Solid-State Circuits Conference-(ISSCC) Digest of Technical Papers*, pages 1–3. IEEE, 2015.
- [35] Hao Luo, Yan Han, Ray CC Cheung, Xiaopeng Liu, and Tianlin Cao. A 0.8-v 230- μw 98-db dr inverter-based $\sigma\delta$ modulator for audio applications. *IEEE Journal of Solid-State Circuits*, 48(10):2430–2441, 2013.
- [36] Hasham Ahmed Khushk, Chadi Jabbour, Patrick Loumeau, et al. High pass filter implementation comparison in unity stf high pass $\delta\sigma$ modulator. In *2011 18th IEEE International Conference on Electronics, Circuits, and Systems*, pages 101–104. IEEE, 2011.

- [37] Orla Feely and Leon O Chua. The effect of integrator leak in sigma-delta modulation. *IEEE Transactions on circuits and systems*, 38(11):1293–1305, 1991.
- [38] James Candy. A use of limit cycle oscillations to obtain robust analog-to-digital converters. *IEEE Transactions on Communications*, 22(3):298–305, 1974.
- [39] Pooya Torkzadeh, Armin Tajalli, and M Atarodi. Modeling of non-ideal switch-capacitor integrator and its effect on baseband sigma-delta modulator stability and output dynamic range. In *2005 International Conference on Microelectronics*, pages 4–pp. IEEE, 2005.
- [40] Piero Malcovati, Simona Brigati, Fabrizio Francesconi, Franco Maloberti, Paolo Cusinato, and Andrea Baschirotto. Behavioral modeling of switched-capacitor sigma-delta modulators. *IEEE Transactions on Circuits and Systems I: Fundamental theory and applications*, 50(3):352–364, 2003.
- [41] Dirk Weiler, T Van Den Boom, and B Hosticka. Resolution prediction for bandpass-sd-modulators using simulink behavior simulation. In *IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS*, number 1, pages I–309. IEEE; 1999, 2003.
- [42] Hashem Zare-Hoseini, Izzet Kale, and Omid Shoaie. Modeling of switched-capacitor delta-sigma modulators in simulink. *IEEE transactions on instrumentation and measurement*, 54(4):1646–1654, 2005.
- [43] Adrian Leuciuc and Cristian Mitrea. On the effect of op-amp finite gain in delta-sigma modulators. In *2000 IEEE International Symposium on Circuits and Systems. Emerging Technologies for the 21st Century. Proceedings (IEEE Cat No. 00CH36353)*, volume 3, pages 754–757. IEEE, 2000.
- [44] WJ Wolski. Influence of op-amp nonidealities and switch-on resistances on sc integrator. In *1988., IEEE International Symposium on Circuits and Systems*, pages 1995–1998. IEEE, 1988.
- [45] Mohammad Honarparvar, Mona Safi-Harb, and Mohamad Sawan. An amplifier-shared inverter-based mash structure $\delta \sigma$ modulator for smart sensor interfaces. In *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2250–2253. IEEE, 2016.
- [46] Zhenglin Yang, Libin Yao, and Yong Lian. A 0.5-v 35- μ w 85-db dr double-sampled $\delta \sigma$ modulator for audio applications. *IEEE Journal of solid-state circuits*, 47(3):722–735, 2012.

- [47] Frank M Yaul and Anantha P Chandrakasan. A noise-efficient $36 \text{ nV}/\sqrt{\text{surd}} \text{ hz}$ chopper amplifier using an inverter-based 0.2-v supply input stage. *IEEE Journal of Solid-State Circuits*, 52(11):3032–3042, 2017.
- [48] Lishan Lv, Xiong Zhou, Zhiliang Qiao, and Qiang Li. Inverter-based subthreshold amplifier techniques and their application in $0.3\text{-v} \delta \sigma$ -modulators. *IEEE Journal of Solid-State Circuits*, 54(5):1436–1445, 2019.
- [49] Clemens M Zierhofer. Analysis of a switched-capacitor second-order delta-sigma modulator using integrator multiplexing. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 53(8):787–791, 2006.
- [50] J Riches, N Erdol, and B Alhalabi. Statistical error shaping for mismatch cancellation in complex bandpass delta-sigma modulators. In *2000 IEEE Workshop on SIGNAL PROCESSING SYSTEMS. SiPS 2000. Design and Implementation (Cat. No. 00TH8528)*, pages 771–780. IEEE, 2000.
- [51] Pooya Torkzadeh and Mojtaba Atarodi. Channel charge injection analysis and its modeling in z-domain for switched-capacitor integrators. In *2009 52nd IEEE International Midwest Symposium on Circuits and Systems*, pages 126–129. IEEE, 2009.
- [52] Yong-Ping Xu, Graham Rigby, and Matthew Starr. Effect of switch charge injection on/spl sigma//spl delta/modulator. In *1995 IEEE TENCON. IEEE Region 10 International Conference on Microelectronics and VLSI.'Asia-Pacific Microelectronics 2000'. Proceedings*, pages 131–134. IEEE, 1995.
- [53] R van de Plassche. A sigma-delta modulator as an a/d converter. *IEEE Transactions on Circuits and Systems*, 25(7):510–514, 1978.
- [54] Jacques Robert, Gabor C Temes, Vlado Valencic, Roger Dessoulavy, and Philippe Deval. A 16-bit low-voltage cmos a/d converter. *IEEE Journal of Solid-State Circuits*, 22(2):157–163, 1987.
- [55] RUDOLF Koch, BERND Heise, FRANZ Eckbauer, EDUARD Engelhardt, JOHN A Fisher, and Franz Parzefall. A 12-bit sigma-delta analog-to-digital converter with a 15-mhz clock rate. *IEEE Journal of Solid-State Circuits*, 21(6):1003–1010, 1986.
- [56] James Candy. A use of double integration in sigma delta modulation. *IEEE transactions on communications*, 33(3):249–258, 1985.

- [57] Tapani Ritoniemi, Teppo Karema, and Hannu Tenhunen. Design of stable high order 1-bit sigma-delta modulators. In *IEEE International Symposium on Circuits and Systems*, pages 3267–3270. IEEE, 1990.
- [58] János Márkus. Higher-order incremental delta-sigma analog-to-digital converters. 2005.
- [59] Qiang Fu, Jian Yang, Liang Liu, Pengfei Wang, Weiping Chen, and Xiaowei Liu. Design of high-performance fourth-order single-loop sigma-delta modulator for inertial sensor. In *2012 International Conference on Optoelectronics and Microelectronics*, pages 461–465. IEEE, 2012.
- [60] Jacques Robert and Philippe Deval. A second-order high-resolution incremental a/d converter with offset and charge injection compensation. *IEEE Journal of Solid-State Circuits*, 23(3):736–741, 1988.
- [61] Vincent Quiquempoix, Philippe Deval, Alexandre Barreto, Gabriele Bellini, János Márkus, José Silva, and Gabor C Temes. A low-power 22-bit incremental adc. *IEEE Journal of Solid-State Circuits*, 41(7):1562–1571, 2006.
- [62] Ali Agah, Katelijn Vleugels, Peter B Griffin, Mostafa Ronaghi, James D Plummer, and Bruce A Wooley. A high-resolution low-power oversampling adc with extended-range for bio-sensor arrays. In *2007 IEEE Symposium on VLSI Circuits*, pages 244–245. IEEE, 2007.
- [63] Ting-Yang Wang and Tai-Cheng Lee. A 84.7-dr wide bw incremental adc using ct structure. In *VLSI Design, Automation and Test (VLSI-DAT)*, pages 1–4. IEEE, 2015.
- [64] Yu-Lun Hsieh and Tai-Cheng Lee. A sar-assisted continuous-time incremental $\sigma\delta$ adc with first-order noise coupling. In *2019 IEEE 13th International Conference on ASIC (ASICON)*, pages 1–4. IEEE, 2019.
- [65] Guangyi Chen You You, Wengao Lu, Xueyou Shi, Zhongjian Chen, and Yacong Zhang. A 16-bit 64-channel 2-order incremental $\sigma\delta$ adc with on-chip decimator for x-ray detections. In *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2019.
- [66] Sha Tao and Ana Rusu. A power-efficient continuous-time incremental sigma-delta adc for neural recording systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 62(6):1489–1498, 2015.
- [67] Sung-En Hsieh and Chih-Cheng Hsieh. A 0.4-v 13-bit 270-ks/s sar-isdms adc with opamp-less time-domain integrator. *IEEE Journal of Solid-State Circuits*, 54(6):1648–1656, 2019.

- [68] Dong Wu, Gencen Gao, Hui Liu, and Nan Xie. A low power double-sampling extended counting adc with class-ab ota for sensor arrays. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 62(1):29–38, 2014.
- [69] John McNeill, Michael CW Coln, and Brian J Larivee. "split adc" architecture for deterministic digital background calibration of a 16-bit 1-ms/s adc. *IEEE Journal of Solid-State Circuits*, 40(12):2437–2445, 2005.
- [70] Imran Ahmed and David A Johns. An 11-bit 45 ms/s pipelined adc with rapid calibration of dac errors in a multibit pipeline stage. *IEEE Journal of Solid-State Circuits*, 43(7):1626–1637, 2008.
- [71] A. Gharbiya and D. A. Johns. On the implementation of input-feedforward delta-sigma modulators. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 53(6):453–457, 2006.
- [72] Y. Fujimoto, Y. Kanazawa, P. Lo Re, and K. Iizuka. A 100 ms/s 4 mhz bandwidth 70 db snr $\delta\sigma$ adc in 90 nm cmos. *IEEE Journal of Solid-State Circuits*, 44(6):1697–1708, 2009.
- [73] K. Lee and G. C. Temes. Improved architecture for low-distortion $\delta\sigma$ adc's. *Electronics Letters*, 45(14):730–731, 2009.
- [74] J. Silva, U. Moon, J. Steensgaard, and G. C. Temes. Wideband low-distortion delta-sigma adc topology. *Electronics Letters*, 37(12):737–738, 2001.
- [75] S. Lee and Y. Chiu. Digital calibration of nonlinear memory errors in sigma–delta modulators. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 57(9):2462–2475, 2010.
- [76] Alireza Bafandeh and Mohammad Yavari. Digital calibration of amplifier finite dc gain and gain bandwidth in mash $\sigma\delta$ modulators. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 63(4):321–325, 2015.
- [77] Tao Sun, A. Wiesbauer, and G. C. Temes. Adaptive compensation of analog circuit imperfections for cascaded delta-sigma analog-to-digital converters. In *Proceedings of the 24th European Solid-State Circuits Conference*, pages 408–411, 1998.
- [78] Peter Kiss, Jose Silva, Andreas Wiesbauer, Tao Sun, Un-Ku Moon, John T Stonick, and Gabor C Temes. Adaptive digital correction of analog errors in mash adcs. ii. correction using test-signal injection. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 47(7):629–638, 2000.

- [79] K. A. O'Donoghue, P. J. Hurst, and S. H. Lewis. A digitally calibrated 5-mw 2-ms/s 4th-order $\Delta\Sigma$ adc in 0.25- μm cmos with 94 db sfdr. In *2010 Proceedings of ESSCIRC*, pages 422–425, 2010.
- [80] D. Kim, T. Matsuura, and B. Murmann. A continuous-time, jitter insensitive $\sigma\delta$ modulator using a digitally linearized gm-c integrator with embedded sc feedback dac. In *2011 Symposium on VLSI Circuits - Digest of Technical Papers*, pages 38–39, 2011.
- [81] János Márkus, José Silva, and Gabor C Temes. Theory and applications of incremental/spl delta/spl sigma/converters. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 51(4):678–690, 2004.
- [82] Jesper Steensgaard, Zhiqing Zhang, Wenhuan Yu, Attila Sárhegyi, Luca Lucchese, Dae-Ik Kim, and Gabor C Temes. Noise–power optimization of incremental data converters. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 55(5):1289–1296, 2008.
- [83] Sam Kavusi, Hossein Kakavand, and Abbas El Gamal. On incremental sigma-delta modulation with optimal filtering. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 53(5):1004–1015, 2006.
- [84] Sylvain Maréchal, Francois Krummenacher, and Maker Kayal. Optimal filtering of an incremental second-order mash11 sigma-delta modulator. In *2011 18th IEEE International Conference on Electronics, Circuits, and Systems*, pages 240–243. IEEE, 2011.
- [85] K. L. Chan, N. Rakuljic, and I. Galton. Segmented dynamic element matching for high-resolution digital-to-analog conversion. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 55(11):3383–3392, 2008.
- [86] Dragos George Ducu and Anca Manolescu. A 2-1 cascaded hybrid continuous-discrete time sigma-delta modulator. In *Proceedings of the 2014 6th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pages 27–30. IEEE, 2014.
- [87] Li Huang, Caroline Lelandais-Perrault, Anthony Kolar, and Philippe Benabes. A new algorithm for an incremental sigma-delta converter reconstruction filter. In *Proceedings of the 32nd Symposium on Integrated Circuits and Systems Design*, pages 1–6, 2019.
- [88] KN Abhilash, Shakthi Bose, and Anu Gupta. A high gain, high cmrr two-stage fully differential amplifier using gm/i d technique for bio-medical applications. In *2013 IEEE Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia)*, pages 40–45. IEEE, 2013.

- [89] Youngcheol Chae, Minho Kwon, and Gunhee Han. A 0.8-/spl mu/w switched-capacitor sigma-delta modulator using a class-c inverter. In *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No. 04CH37512)*, volume 1, pages 1–1152. IEEE, 2004.
- [90] Mohamed Atef, Hong Chen, and Horst Zimmermann. 10gb/s inverter based cascode transimpedance amplifier in 40nm cmos technology. In *2013 IEEE 16th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)*, pages 72–75. IEEE, 2013.
- [91] Hao Luo, Yan Han, Ray CC Cheung, Xiaopeng Liu, and Tianlin Cao. A 0.8-v 230-mu w 98-db dr inverter-based $\sigma\delta$ modulator for audio applications. *IEEE Journal of Solid-State Circuits*, 48(10):2430–2441, 2013.
- [92] Pierre Bisiaux. *Etude et conception de CAN haute résolution pour le domaine de l'imagerie*. PhD thesis, 2018.
- [93] Pooya Torkzadeh, Armin Tajalli, and M Atarodi. Modeling of non-ideal switch-capacitor integrator and its effect on baseband sigma-delta modulator stability and output dynamic range. In *2005 International Conference on Microelectronics*, pages 4–pp. IEEE, 2005.
- [94] V Georgiev and K Stantchev. Adequate sc model of op-amp with finite dc gain and bandwidth suitable for two-op-amp sc biquad analysis. *Electronics Letters*, 22(19):996–997, 1986.
- [95] Hashem Zare-Hoseini and Izzet Kale. On the effects of finite and nonlinear dc-gain of the amplifiers in switched-capacitor/spl delta//spl sigma/modulators. In *2005 IEEE International Symposium on Circuits and Systems*, pages 2547–2550. IEEE, 2005.

Titre: Conception et calibration d'un banc de convertisseurs analogique-numérique Sigma-Delta incrémental « two-step »

Mots clés: Modulateur Sigma-Delta Incrémental, Calibration Numérique, Filtre de Reconstruction, Amplificateur à Inverseur

Résumé: Dans le cadre des imageurs Haute Définition, une tendance est d'intégrer un banc de convertisseurs analogiques-numériques jouxtant la matrice de pixel. La contrepartie est une contrainte sur le facteur de forme du convertisseur. Un convertisseur de type Sigma-Delta incrémental à base d'inverseur a été conçu lors de travaux précédents en respectant ces contraintes. Mais le placement-routage du circuit a abouti à une dégradation des performances à savoir une résolution de 9 bits au lieu des 14 bits escomptés. Une méthode de calibration s'imposait donc. Cette thèse propose plusieurs méthodes de correction implémentées par des filtres numériques appliqués sur les bits de sortie

et sur des combinaisons des bits de sorties pour tenir compte de phénomènes non-linéaires observés en simulation « post-placement-routage ». Les méthodes ont été validées à partir des résultats de simulation « post-placement-routage » et permettent d'atteindre 14 bits de résolution. Pour aller plus loin, la thèse propose également un modèle des défauts du circuit au niveau des intégrateurs qui sont la partie la plus critique du circuit. Ce modèle, qui met en œuvre des capacités parasites, rejoint les résultats de simulation « post-placement-routage » avec une très bonne précision ce qui permet d'envisager des voies d'amélioration pour une prochaine conception.

Title: Calibration of a two-step Incremental Sigma-Delta Analog-to-Digital Converter

Keywords: Incremental Sigma-Delta Modulator, Digital Calibration, Reconstruction Filtre, Inverter-based Amplifier

Abstract: In the context of High Definition imagers, a trend is to integrate a bank of analog-to-digital converters adjacent to the pixel matrix. The disadvantage is a constraint on the form factor of the converter. An incremental inverter-based Sigma-Delta converter was designed during previous work while respecting these constraints. But the post-layout of the circuit resulted in a performance degradation namely a resolution of 9 bits instead of the expected 14 bits. A calibration method was therefore necessary.

ods implemented by digital filters applied on the output bits and on combinations of the output bits to take account of non-linear phenomena observed in post-layout simulation. The methods have been validated from the post-layout simulation results and achieve 14-bit resolution. To go further, the thesis also proposes a model of the circuit defects at the level of the integrators which are the most critical part of the circuit. This model, which implements parasitic capacitances, joins the post-layout simulation results with a very high precision, which makes it possible to consider ways of improvement for a future design.

This thesis proposes several correction meth-